UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

**Diseño y Evaluación de Técnicas de Reconocimiento de
Idioma mediante la Fusión de Información Fonotáctica y
Acústica**

TESIS DOCTORAL

Autor:

Ing. Christian Salamea Palacios

2018

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

**Diseño y Evaluación de Técnicas de Reconocimiento de
Idioma mediante la Fusión de Información Fonotáctica y
Acústica**

Tesis Doctoral

Autor:

Christian Salamea Palacios

Ingeniero Electrónico

Universidad Politécnica Salesiana del Ecuador

Directores:

Ricardo de Córdoba Herralde

Doctor Ingeniero de Telecomunicación

Catedrático del Departamento de Ingeniería Electrónica

Universidad Politécnica de Madrid

Luis Fernando D'Haro Enriquez

Doctor Ingeniero de Telecomunicación

Research Scientist::Human Language Technologies

Institute for Infocomm Research (I2R - A*STAR)

2018

TESIS DOCTORAL

Diseño y Evaluación de Técnicas de Reconocimiento de Idioma mediante la Fusión de Información Fonotáctica y Acústica

Autor: Christian Salamea Palacios

Directores: Ricardo de Córdoba Herralde y Luis Fernando D'Haro Enriquez

Tribunal nombrado por el Mgfc. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día de de 2018.

Presidente: D.....

Vocal: D.....

Vocal: D.....

Vocal: D.....

Secretario: D.....

Realizado el acto de defensa y lectura de la Tesis el día.....de..... de 2018 en.....

Calificación:.....

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

Agradecimientos

Gracias a la gente del GTH por permitirme ser parte de su mundo, por darme la posibilidad de ver y aprender una manera diferente de enfocar la ciencia, la cultura, la educación y la comunicación misma. Todo lo asimilado en estos años, de seguro lo podré transmitir y volcar en mi entorno donde será de mucha utilidad.

Gracias también a la gente de la UPS en Ecuador, en especial al Padre Javier Herrán y a Luis Tobar que siempre han estado pendientes de mis avances y dispuestos a brindarme todo el apoyo.

A los dos grandes directores de este trabajo, que han aportado a que este trabajo lleve una calidad extraordinaria. Gracias por vuestro tiempo y vuestra sabiduría. A mis compañeros de despacho, Jaime, Verónica y particularmente a Bea con quien he compartido esta etapa final llegando a descubrir a la gran persona que es y a una amiga inmensa.

A mis padres, quienes a pesar de la dificultades que representa la distancia han sabido ser cercanos, tanto que no he llegado a sentir la soledad propia de estar en un entorno diferente. A mis hermanas que me han ofrecido una ayuda infinita para mantener la conexión con Ecuador, por entenderme y darme la confianza de que siempre podía contar con alguien a la distancia.

A Nacho, gracias por existir hijo, gracias por ser quien eres. Desde el inicio has sido mi fuerza, mi motor y mi razón de existencia. Le doy gracias a Dios por haberme dado la oportunidad de compartir contigo lo que soy y lo que creo de la vida. Gracias por todo lo que me aportas, por transportarme a un plano más puro y natural. A Cristina, gracias por no haber obstaculizado en ningún momento esta aventura, en su inicio, en su desarrollo ni en su parte final.

A Rick, por estar siempre ahí para ofrecer una orientación, un consejo, una alternativa, una solución o cualquier tipo de apoyo. Por darme la oportunidad de conocer a la persona más allá del Director y permitiéndome descubrir a un gran amigo. Gracias por los consejos, la paciencia y la guía durante todo este tiempo. A día de hoy, intento brindar ese mismo apoyo a la gente que ahora depende de mí. Es invaluable todo lo que he podido aprender de tu experiencia.

Un agradecimiento muy especial y cariñoso a aquellos ángeles que han aparecido en mi vida durante este tiempo, siendo una bendición para mí, enseñándome como disfrutar del vivir, a sentirme pleno y único. Por transformar a este lustro en el lustro ganado, el lustro que ha cambiado por completo a la persona que llegó y la que se va.

Gracias a la vida misma, por la experiencia ganada y la interminable posibilidad de seguir aprendiendo.

Resumen

La aplicación de técnicas fonotácticas en los sistemas de reconocimiento de idioma ha venido siendo un ámbito de continuo estudio ya que su correcta utilización deriva en importantes mejoras en el rendimiento de dichos sistemas. La forma en la que se desarrolla un idioma así como el conjunto de características fonéticas que se generan con el habla son elementos claves en las tareas de identificación de idioma (LID). La eficiencia con la que se logran capturar dichas características fonéticas es un factor determinante para la obtención de un reconocedor de calidad. Aunque los sistemas actuales han alcanzado una tasa de acierto muy razonable, continúan teniendo problemas, como por ejemplo la cantidad de recursos informáticos requeridos para el procesamiento de la información y por otro lado la cantidad de información de entrenamiento necesaria para que los sistemas automáticos puedan incorporar adecuadamente información característica de los idiomas a reconocer. Las redes neuronales profundas y particularmente las recurrentes, han resultado eficientes para modelar las características fonéticas de los idiomas y por tanto, se están utilizando con este fin para varios tipos de tareas en el reconocimiento de habla y en tareas de LID. Los modelos de lenguaje se generan a dos niveles, uno a nivel *léxico* y otro a nivel *fonético*. En esta tesis se ha decidido utilizar un sistema fonotáctico que es capaz de aprovechar una mayor información de contexto y para ello, se han utilizado unidades fonéticas que buscan incorporar las características fonotácticas de idioma, además de incorporar más información de contexto de la que ofrece un fonema. En esta tesis se explora el uso de estas unidades fonéticas *ngramas-fonéticos* en tareas LID, identificando valores óptimos de configuración y respuestas mediante las diferentes técnicas propuestas, todo ello en el contexto de la creación de modelos de lenguaje basados en redes neuronales recurrentes. Por otra parte, en el mismo ámbito fonotáctico, se introduce la idea de utilizar la representación vectorial de *ngramas-fonéticos* en tareas LID, dejando de lado el concepto de modelo de lenguaje que se basa en información del pasado para predecir nueva información y dando paso a la generación de modelos basados en el contexto y en los *ngrama-fonéticos* objetivo. También se han estudiado en esta tesis los sistemas fonotácticos para tareas LID basados en estructuras de i-Vectores. El uso de información discriminativa y de coeficientes PLLR han permitido explorar nuevas alternativas en la tarea LID. En base a ello, se han estudiado alternativas para ampliar el contexto que tienen en cuenta dichos coeficientes para mejorar su rendimiento. Todos los estudios propuestos han sido realizados sobre la base de datos KALAKA-3 utilizada en la evaluación ALBAYZIN-LRE2012 en razón del equilibrio encontrado entre su tamaño y la dificultad de la tarea definida en cuanto a su tiempo de ejecución.

Abstract

The application of phonotactic techniques in language recognition systems has always been an area of special interest since, if correctly used, it leads to significant improvements in the performance of recognition systems. The acoustic realization of the language and its phonetic characteristics are the key elements for the language recognition task (LID). The efficiency obtained with these phonetic characteristics determines the quality of the recognizer. Although nowadays the efficiency of the recognizers is very high, there are still several problems remaining, e.g., they use high computational resources to process the information and, also, the training data is always not enough to incorporate all the characteristics specific of a language.

Deep Neural Networks and especially the recurrent ones, have proved to be efficient to model the phonetic characteristics of the languages and, so, they are being used for several tasks in speech recognition and speaker/language identification.

Language models are generated in two levels, either a lexical level or a phonetic level. In this thesis, we have decided to use a phonotactic system that is able to manage a larger context information and, to that end, we propose the use of the *phonetic ngram*, that tries to incorporate the phonotactic characteristics of the languages, together with the context information that phonemes alone do not provide.

In this thesis, we explore the use of these phonetic features in LID tasks, finding optimum values for the configuration parameters and presenting different techniques, all of them related in the creation of language models based on recurrent neural networks.

On the other hand, using also a phonotactic approach, we introduce the idea of using the vector representation of phonetic ngrams for LID tasks, as an alternative to the language model based on RNN, to create models based in the context and the objective phonetic ngrams.

We have also studied in this thesis the phonotactic systems based on i-Vectors for LID tasks. The use of discriminative information and the PLLR coefficients have offered us new alternatives in the LID task. We have proposed alternatives to increase the context considered in these parameters to improve its performance.

All of these studies have been applied to the KALAKA-3 database used in the ALBAYZIN-LRE2012 evaluation, because of the good compromise between the size of the database and the task difficulty in relation with its execution time.

Índice general

Agradecimientos	III
Resumen	I
Índice de figuras	VII
Índice de cuadros	IX
Lista de Abreviaturas	XI
1. INTRODUCCIÓN	1
1.1. Motivación	1
1.2. Objetivos	6
1.3. Organización	8
2. ESTADO DE LA CUESTIÓN	11
2.1. La identificación de idioma y su clasificación	11
2.1.1. Generalidades	11
2.1.2. Sistemas LID acústicos	13
2.1.3. Sistemas LID lingüísticos	14
Reconocedores fonéticos en paralelo seguidos de modelos de len- guaje (PPRLM)	15
2.2. Modelos de lenguaje basados en redes neuronales recurrentes usando ngramas- fonéticos (RNNLM-P)	17
2.3. El modelado en el espacio vectorial	21
2.3.1. Descripción	21
2.3.2. Modelo Skip-Gram	23
2.3.3. Negative Sampling (NS)	24
2.3.4. Modelos GloVe	25
2.4. Los i-Vectores y la información discriminativa entre idiomas como pará- metros en tareas LID	26
2.4.1. Supervectores	26
2.4.2. Joint Factor Analysis	27
2.4.3. I-Vectores	29
2.4.4. Modelado en el Subespacio Multinomial	30
2.4.5. Categorización basada en n-gramas	32
2.5. Parámetros PLLR y Coeficientes Cepstrales con información temporal . .	32
2.5.1. Parámetros PLLR	32
2.5.2. Coeficientes Cepstrales Delta Desplazados (SDCs)	33

2.6.	Fusión, Calibración y Regresión Logística Multi-clase	34
2.6.1.	Fusión de Información en LID	34
	Método de combinación lineal para la fusión de información	35
	Método de combinación logarítmica para la fusión de información	35
2.6.2.	Calibración	36
2.6.3.	Regresión Logística Multi-clase	38
3.	MARCO DE EXPERIMENTACIÓN	41
3.1.	Bases de datos	41
3.1.1.	KALAKA-3	41
3.2.	Métricas utilizadas para definir el rendimiento de los sistemas de recono- cimiento	42
3.2.1.	Entropía	42
3.2.2.	Función de coste de detección promedio (Cavg)	43
3.3.	Los reconocedores de fonemas	43
3.4.	Sistema acústico de referencia	44
4.	POSTERIORGRAMAS E INFORMACION DISCRIMINATIVA	47
4.1.	Introducción	47
4.2.	Sistema LID basado en posteriorgramas	48
4.3.	Sistema de rankings de información discriminativa	50
4.4.	Incorporación de información discriminativa en los posteriorgramas	52
4.4.1.	Factor de suavizamiento en los posteriorgramas discriminativos	55
4.4.2.	Resultados para bigramas	55
4.4.3.	Resultados para trigramas	56
4.4.4.	Rankings discriminativos y el umbral mínimo de ocurrencia de los n-gramas	56
4.4.5.	Fusión de los posteriorgramas discriminativos y el sistema acústico de referencia	57
4.4.6.	Conclusiones y aportaciones	58
5.	PARÁMETROS SDPCs PARA TAREAS LID	59
5.1.	Introducción	59
5.2.	Sistema de LID basado en Shifted Delta PLLR Coefficients (SDPCs)	60
5.2.1.	Descripción	60
5.2.2.	Sistema de partida basado en los parámetros SDPCs	61
5.2.3.	Modificación del parámetro D	64
5.2.4.	Modificación del parámetro K	64
5.2.5.	Modificación del parámetro P	65
5.2.6.	Conclusiones y aportaciones	67
6.	APLICACIÓN DE REDES NEURONALES RECURRENTE A SISTEMAS LID	69
6.1.	Descripción de la técnica usada para incorporar ngramas-fonéticos en las RNNs	69
6.2.	Estructura de la experimentación	72
6.2.1.	Descripción de los parámetros configurables de la RNN	74
	Número de neuronas en la capa de estado (NNE)	74

Número de clases (NCS)	75
Memoria de la RNN (MEM)	77
6.2.2. Configuración de la red neuronal	77
6.3. Efectos del uso de ngramas-fonéticos en la generación de RNNLMs	79
6.3.1. Número de iteraciones al aumentar el orden del n-grama	80
6.3.2. Para 1gramas	81
6.3.3. Para 2gramas-fonéticos	82
6.3.4. Para 3gramas-fonéticos	84
6.3.5. Fusión de los sistemas RNNLM-P, PPRLM Y Acústico	88
6.3.6. Conclusiones y aportaciones	90
7. APLICACIÓN DE EMBEDDINGS NEURONALES EN TAREAS LID	91
7.1. Creación de embeddings neuronales fonotácticos	91
7.2. Embeddings Neuronales usados como vectores de parámetros	93
7.2.1. Descripción del sistema	93
Embeddings con información de contexto	95
7.2.2. Configuración del sistema	96
7.2.3. Resultados	96
Embedding único agrupado	96
Embedding individual	97
Utilización de GloVe para generar los ENs	99
7.2.4. Fusión con el sistema acústico basado en MFCCs	100
7.3. Aplicación de Embeddings neuronales para optimizar las secuencias de ngramas-fonéticos	101
7.3.1. Descripción del sistema	101
Clustering con Embeddings Neuronales	102
Sustitución de unidades menos frecuentes usando embeddings y reglas heurísticas	102
7.3.2. Configuración del sistema	103
7.3.3. Resultados	104
Aplicación del Clustering con Embeddings Neuronales	104
Aplicación de la sustitución de unidades menos frecuentes usando embeddings y reglas heurísticas	105
7.3.4. Fusión con el sistema acústico basado en MFCCs	105
7.3.5. Conclusiones y aportaciones	106
8. CONCLUSIONES, LINEAS FUTURAS Y PUBLICACIONES	109
8.1. Conclusiones	109
8.2. Líneas futuras	113
8.3. Publicaciones	114
8.3.1. Publicaciones en Revistas Internacionales:	114
8.3.2. Publicaciones en Revistas Nacionales:	114
8.3.3. Conferencias/Congresos:	114
Bibliografía	117

Índice de figuras

2.1. Sistema LID básico	13
2.2. Niveles de Información usados por LID	15
2.3. Estructura de un sistema PPRLM para tareas LID	16
2.4. Red neuronal de una capa de entrada y una de estado	18
2.5. Generación de los ngramas-fonéticos a partir de una secuencia de fonemas	20
2.6. Modelo Skip-Gram	24
2.7. Cálculo del SDC para una trama de tiempo T	34
2.8. Fusión a nivel de parámetros de entrada	35
2.9. Fusión a nivel de scores a la salida de los módulos Back-End	35
2.10. Sistemas LID desde el punto de vista de la calibración	37
3.1. Estructura del sistema acústico basado en MFCCs	46
4.1. Sistema Fonotáctico basado la obtención de i-Vectores	48
4.2. Procedimiento de creación de los posteriorgramas	49
4.3. Incorporación del ranking discriminativo a los posteriorgramas	54
5.1. Sistema SDPC basado en i-Vectores	60
5.2. Obtención de los coeficientes SDPCs	62
6.1. Arquitectura típica de una Red Neuronal Recurrente	71
6.2. Sistema de reconocimiento de idioma	72
6.3. Sistema LID basado en RNNLMs	74
6.4. Factorización de la capa de salida	76
6.5. MEM en Redes Neuronales Recurrentes	78
6.6. Efectos en la perplejidad para el conjunto de evaluación al modificar NNE	78
6.7. Resultados al modificar MEM	79
6.8. Cavg de cada uno de los HMMs utilizando 1gramas	82
6.9. Cavg con parámetros óptimos para 1gramas	82
6.10. Resultados variando la factorización de la capa de salida para 2gramas- fonéticos en la generación de RNNLMs	83
6.11. Cavg óptimos de 1gramas y 2gramas con MEM variable	84
6.12. Óptimo de Cavg con 3gramas-fonéticos variando MEM	85
6.13. Resultados de modificar k en la aplicación del umbral	87
6.14. Comparación del Cavg aplicando o no el umbral	87
6.15. Cavg óptimo para 1gramas, 2gramas y 3gramas-fonéticos	88
7.1. Ejemplo de un vector de embedding de un ngrama-fonético	92
7.2. Obtención de los embeddings neuronales	92

7.3. Secuencias de ngramas-fonéticos usadas como vectores de parámetros en LID	94
7.4. Sistema LID por medio de embeddings neuronales	95
7.5. Formación de embeddings de contexto	96
7.6. Sistema de reconocimiento basado en RNNLMs con ENs	103

Índice de cuadros

3.1. Estadísticas de la base de datos KALAKA-3	42
3.2. Sistema Acústico de referencia con Coeficientes MFCCs para KALAKA-3.	46
4.1. Sistema Fonotáctico de referencia basado en i-Vectores usando posterior-gramas.	50
4.2. Formato del ranking utilizado.	52
4.3. Efectos de incluir información discriminativa en el sistema basado en posteriorgramas.	54
4.4. Factor de suavizamiento para bigramas.	56
4.5. Factor de suavizamiento para trigramas.	56
4.6. Cambio en el umbral mínimo de ocurrencia para trigramas.	57
4.7. Resultados de fusionar el módulo acústico con el sistema fonotáctico de referencia y el basado en información discriminativa.	57
5.1. Configuración de partida para el sistema SDPC de fonemas y estados.	64
5.2. Configuraciones óptimas del SDC en la creación de los SDPCs.	66
5.3. C_{avg} para cada una de las configuraciones - Caso fonemas	66
5.4. C_{avg} para cada una de las configuraciones - Caso estados	66
5.5. Fusión del sistema SDPCs con el sistema acústico basado en MFCCs.	67
6.1. Parámetros óptimos de la RNN usando 1-grams en la base de datos KALAKA-3.	79
6.2. Número de ngramas-fonéticos por cada modelo del reconocedor fonético.	80
6.3. Valores de mejora mínima en la generación de modelos de lenguaje.	81
6.4. Comportamiento del C_{avg} para 1gramas y 2gramas.	84
6.5. log-probabilidades asignadas por las RNNLMs y por SRI-LM para dos 3-gramas no aparecidos en el entrenamiento.	86
6.6. Fusión de los ngramas-fonéticos de 1ero, 2do y 3er orden.	89
6.7. Comparación de RNNLM-P con PPRLM y Acústico.	89
6.8. Fusiones de los sistemas PPRLM, Acústico y RNNLM-P.	90
7.1. Parámetros óptimos Embedding individual.	98
7.2. C_{avg} para cada idioma y Fusión para embeddings individuales con Skip-Gram y contexto B-A-B (25-50-25).	98
7.3. Comparación de las técnicas <i>Embedding único agrupado</i> y <i>Embedding individual</i>	98
7.4. C_{avg} para cada idioma y su Fusión para embeddings individuales con GloVe y contexto B-A-B (25-50-25).	100

7.5. Sistema Acústico de referencia con Coeficientes MFCCs para KALAKA-3.	100
7.6. Elementos alofónicos dobles.	103
7.7. Aplicación del clustering con embeddings	105
7.8. Aplicación de la sustitución de unidades frecuentes	106
7.9. Fusión del sistema Embedding <i>Reemplazo menos frecuentes con Acústico.</i> .	106
8.1. Resultados óptimos de las técnicas propuestas.	112
8.2. Fusión de las diferentes técnicas propuestas <i>Fusión de las diferentes técnicas propuestas</i>	113

Lista de Abreviaturas

BF	B ottleneck F eatures
BPTT	B ack P ropagation T hrough T ime
Cavg	C ost A verage F unction
DNNs	D eep N eural N etworks
ENs	E mbeding N euronales
GMMs	G aussian M ulti-dimentional M odelling
HMMs	H idden M arkov M odels
ID	I nformación D iscriminativa
JFA	J oint F actor A nalysis
LDA	L inear D iscriminant A nalysis
LID	L anguage I dentification
LSA	L atent S emantic A nalysis
MAP	M aximum A Posteriori
MEM	M emoria de la RNN
MFCCs	M el F requency C epstral C oefficients
MLR	M ulti-class L ogistic R egression
NAP	N uisance A tttribute P rojection
NCS	N úmero de C lases en la capa de S alida
NNE	N úmero de N euronas en la capa de E stado
NNLM	N eural N etwork L anguage M odel
NNs	N eural N etworks
PCA	P rincipal C omponent A nalysis
PLLR	P hone L og L ikelihood R atios
PLPs	P erceptual L inear P redictive C oefficients
PPRLM	P aralell P hone R ecognition L anguage M odelling
RBs	R adial F unctions
RNN	R ecurrent N eural N etwork
RNNLM-P	M odelos de L enguaje basados en RNN y n -gramas fonéticos
SDC	S hifted D elta C oefficients
SDPC	S hifted D elta P hone C oefficients
SMM	S ubspace M ultinomial M odel
SVM	S upport V ector M achine
UBM	U niversal B ackground M odel
VAD	V oice A ctivity D etector
VQ	V ectorial Q uatification
VSM	V ector S ubspace M odelling
1gramas	U nigramas
2grams	2 gram-fonéticos
3grams	3 gram-fonéticos

Dedicado a ...Nacho

Capítulo 1

INTRODUCCIÓN

1.1. Motivación

La comunicación por medio del lenguaje ha venido siendo, a través de los siglos, una de las principales herramientas utilizadas por la humanidad para transmitir ideas, pensamientos y sentimientos, por medio de los cuales ha logrado establecer una convivencia armoniosa. Esta convivencia ha creado las condiciones necesarias para generar un desarrollo cultural, científico y económico de las sociedades en los más diversos ámbitos. En la actualidad existen más de 7 mil millones de personas que lo utilizan en uno o más de los 6912 idiomas conocidos en el mundo (Gordon y Grimes, 2005).

Esta comunicación puede ser comprendida en su forma más básica como un sistema en el que interviene un emisor y un receptor, así como el canal de comunicación. La tecnología de habla, rama de la Inteligencia Artificial, surge con el fin de optimizar mediante diferentes técnicas las condiciones del canal de comunicación (Kanagasundaram y col., 2014) y/o para viabilizar la comunicación entre agentes reales o artificiales en diferentes formas:

- Comunicación hombre-máquina, que se puede ejemplificar en sistemas de transcripción de voz o del habla (secuencias de símbolos representando el habla) (Renals y King, 2010) (Besacier y col., 2014) donde la *máquina* posee la capacidad de identificar palabras expresadas por el *hombre* o en sistemas de identificación de idioma, donde la actuación de la *máquina* junto con la transcripción en sí misma, también adquiere la capacidad de tomar una decisión respecto al idioma al que pertenece una señal de voz evaluada.

- Comunicación máquina-hombre, que se puede ejemplificar en sistemas de síntesis Texto-a-Voz (TTS) donde un texto en particular es reproducido audiblemente mediante diferentes técnicas. La investigación científica en este campo ha llevado a que la síntesis TTS pase de una reproducción plana y monótona a una reproducción dinámica donde se pueden reflejar diferentes sensaciones y emociones (Lorenzo-Trueba y col., 2013).

- Comunicación hombre-máquina-hombre, que se puede ejemplificar en sistemas inteligentes de diálogo. A partir de éstos se han desprendido diversas áreas de estudio, como el *language understanding* (Litman y Silliman, 2004),(Sagae y col., 2009) donde en base a los requerimientos de un usuario, un sistema de diálogo es capaz de generar respuestas guiadas coherentes e inteligentes. Por otro lado, existe el lenguaje de signos, mediante el cual se utilizan a las *máquinas* como una interfaz inteligente de traducción (López-Ludeña y col., 2014).

No son pocas las ocasiones donde, para las diferentes formas de comunicación expuestas resulta prioritario que los actores en un proceso de comunicación (interlocutores) pudiesen entenderse independientemente del idioma materno respectivo. En este contexto, la tarea de identificación automática de idioma (*LID*) con la que se determina el idioma al que se corresponde una señal de voz puede muy bien considerarse como una alternativa viable y segura para conseguir dicho fin. Los ejemplos expuestos por (Muthusamy, Barnard y Cole, 1994) materializan en gran parte la necesidad y la conveniencia de contar con sistemas de identificación de idioma. Entre las principales aplicaciones que actualmente tiene la LID destacan los traductores simultáneos, los call-centers multilingües y los transcriptores de audio. Desde hace más de treinta años están apareciendo nuevas técnicas para LID y siguen evolucionando a la fecha. A continuación se exponen aquellas que actualmente se vienen utilizando y que están relacionadas de una u otra forma con el trabajo desarrollado en esta tesis.

La técnica de Parallel Phone Recognition Language Modelling (Zissman, 1996), (Yan, Barnard y Cole, 1996), consiste en la generación de modelos de lenguaje a partir de secuencias de fonemas obtenidas de diferentes fuentes de información. En el caso que nos ocupa, la señal de voz se decodifica por medio de múltiples reconocedores de fonemas y se genera un conjunto de secuencias fonéticas para cada reconocedor. Estos conjuntos se utilizan luego para entrenar los modelos de lenguaje que serán los que cuantifiquen la probabilidad de que una señal de voz de un idioma desconocido, haya sido expresada en uno u otro de los idiomas previamente entrenados.

Cavnar por otra parte introdujo en la categorización de texto (Cavnar y Trenkle, 1994), la idea de la categorización de las unidades fonéticas de una frase en función de su frecuencia de ocurrencia por medio de una especie de *ranking*. Mediante un método simple y eficaz logró identificar dentro de una fase de entrenamiento a las unidades fonéticas más representativas de cada idioma. De esta forma generó las condiciones necesarias para mejorar el reconocimiento de un texto de idioma desconocido (Cordoba y col., 2007). Sin embargo, la aplicación directa de la propuesta de Cavnar no es óptima ya que las posiciones más altas en el ranking siempre son ocupadas por los n-gramas de menor orden y como se sabe son los menos discriminativos en tareas de LID (D'Haro,

2009). Además, de que en este trabajo de tesis se estudia el comportamiento de los sistemas de reconocimiento de idioma que utilizan el reconocimiento de fonemas como estrategia primaria de tratamiento de la señal de voz, donde la presencia de errores será mayor que los producidos en (Cavnar y Trenkle, 1994) donde la categorización se realiza sobre texto.

D'Haro y Córdoba en (D'Haro, 2009) basados en los resultados obtenidos por (Nagarajan y Murthy, 2006) introdujeron el concepto de discriminabilidad en tareas LID, con el que se otorga mayor relevancia (que se refleja en mejores posiciones en un ranking) a los n-gramas con altas frecuencias de ocurrencia para el idioma a modelar y menor relevancia (peores posiciones en el mismo ranking) a los n-gramas con ninguna o pequeña frecuencia de ocurrencia en los otros idiomas. La información discriminativa obtenida a partir de rankings de frecuencia de n-gramas ha sido utilizada con gran éxito para tareas de LID (Caraballo y col., 2010) y también ha sido usada como elemento complementario de otros sistemas fonotácticos de reconocimiento de idioma (Salamea-Palacios y col., 2013). El efecto discriminativo entre idiomas, además de su aplicación en el entrenamiento de modelos de idiomas (Zhai y col., 2006), (Saraclar y Roark, 2005), o en la selección de parámetros (Penagarikano y col., 2011) también ha sido utilizado en otras tareas relacionadas con el reconocimiento de idioma donde el entrenamiento discriminativo aplicado ha sido utilizado para generar modelos característicos de dialecto (Richardson, Campbell y Torres-Carrasquillo, 2009).

Por otro lado, la separabilidad de las clases a identificar representa un elemento clave en las tareas de reconocimiento, dado que en muchos casos puede resultar prácticamente imposible conseguir dicha separabilidad en el espacio de parámetros original (normalmente de dimensión baja). Sin embargo, dicha separabilidad mejora cuando se utiliza una función que *simula* la conversión del espacio de parámetros original a otro (normalmente de dimensión alta) donde se utiliza un hiperplano multidimensional lineal para obtener separabilidad. Con este fin, se utiliza la técnica de Support Vector Machine (SVM) (Chen, Lin y Schölkopf, 2005) que principalmente utiliza una función kernel para pasar a una dimensión mayor, con lo que las decisiones se pueden tomar en esta última facilitando y optimizando la tarea del clasificador.

Los supervectores (Kinnunen y Li, 2010), (Dominguez, 2011) son elementos esenciales en tareas de reconocimiento. Partiendo de un modelado genérico obtenido a partir de mezclas de gaussianas (UBM), un supervector se define como la concatenación de las medias de dicho modelo adaptado por medio de MAP (Maximum a Posteriori) a la señal de entrada (Reynolds, 2006). A partir de ellos, se ha vuelto común el entrenamiento de i-Vectores que se definen como vectores de baja dimensionalidad que conservan características de dimensionalidad alta (Dehak y col., 2011a) y son ampliamente utilizados

como vectores de parámetros en tareas que involucran tratamiento de habla y en general el tratamiento de señales.

Una de las estrategias que ha permitido disminuir la incertidumbre de los sistemas de reconocimiento a la hora de la toma de decisiones referida al idioma es la combinación de resultados provenientes de diferentes fuentes de información, como por ejemplo la combinación de los resultados provenientes de módulos que utilizan información acústica y fonotáctica (D'Haro y col., 2014). Dada la variabilidad de los resultados obtenidos por los sistemas de reconocimiento de idioma con cada fuente de información, antes de realizar la combinación de los resultados se requiere contar con una tecnología de fusión que sea capaz de homogeneizar la información existente. Han aparecido varias propuestas para lograr este objetivo, entre las más reconocidas está la *calibración* de scores planteada por Brummer, quien propone una metodología basada en el ajuste de pesos en funciones del coste de detección (Brümmer y Preez, 2006). Varios otros autores (Kajarekar y Stolcke, 2007), (Dehak y col., 2009) han propuesto otro enfoque para la *normalización* considerando a la dispersión de datos en el hiperplano vectorial como elemento de análisis. Así, se han definido dos métodos de normalización, los primeros basados en la sesión (que involucra al canal de comunicación) y los segundos basados en los scores (que involucran a cada resultado). La información calibrada y normalizada es susceptible de ser combinada, con el objetivo de extraer las mejores características de los sistemas individuales en un único sistema global, mejorando así la tarea de reconocimiento de idioma en su conjunto.

Los vectores de características utilizados para el reconocimiento son elementos de estudio permanente. Se busca identificar cuáles se adaptan de mejor forma a las condiciones del entorno para que respondan mejor a la variabilidad de las señales a reconocer. En este contexto, en (Vergin, O'shaughnessy y Farhat, 1999) con los Mel Frequency Cepstral Coefficients (MFCCs) y en (Hermansky, 1990) con los Perceptual Linear Predictive Coefficients (PLPs), se exponen metodologías que posibilitan la extracción de información característica de las señales en análisis que es útil para tareas LID. Alrededor de éstas y de otras metodologías se han estudiado variantes que permiten contextualizar aún mejor las señales de entrada. En (Díez y col., 2012), a partir de los MFCCs se propone el uso del logaritmo del cociente de verosimilitud fonética, que se obtiene a partir de las probabilidades a posteriori de los fonemas en cada una de las tramas en las que se ha segmentado la señal de entrada. Debido a los buenos resultados que se obtienen en (Plchot y col., 2014) ó (Diez y col., 2014) se les considera unos parámetros de gran rendimiento para tareas de reconocimiento de idioma.

Las redes neuronales profundas (DNN), (Dahl y col., 2012), (Richardson, Reynolds y Dehak, 2015) y las redes neuronales recurrentes (RNN), (Mikolov y col., 2010) en una

de sus formas, se han utilizado con éxito en diversas aplicaciones de clasificación y reconocimiento, las primeras aprovechando una mejor inicialización de los pesos en los modelos estocásticos, por ejemplo, en el reconocimiento de emociones, ya sea en el ámbito acústico (Stuhlsatz y col., 2011) o en el de texto (Zhang y LeCun, 2015) y las segundas permitiendo modelar de mejor forma las características dinámicas o temporales de las unidades fonéticas que ha de reconocer, por ejemplo, en sistemas de reconocimiento de habla (Mikolov y col., 2011). En el caso de las DNNs, en un principio se utilizaron para tareas de reconocimiento automático de habla (ASR) (Hinton y col., 2012) y ello condujo a su utilización en otras áreas de las tecnologías del habla, como el del reconocimiento de idioma. En este contexto se han venido considerando dos enfoques de estudio. Uno llamado *método directo* en el cual se usa a la DNN entrenada como un clasificador para distinguir entre idiomas y otro llamado *método indirecto* donde se extrae información de la DNN que luego se utiliza para entrenar un segundo clasificador y llevar a cabo la tarea de reconocimiento.

El método directo ha sido propuesto a nivel acústico (Lopez-Moreno y col., 2014) con gran éxito en su comparación con los sistemas clásicos basados en i-Vectores, en tanto que con el método indirecto se han propuesto dos técnicas, las *bottleneck features* (BF) (Matejka y col., 2014) y las *DNN posteriors* (Richardson, Reynolds y Dehak, 2015), y con ambas se ha conseguido mejorar el rendimiento de los sistemas de reconocimiento de habla.

En relación a los métodos indirectos se han propuesto:

- Las BF que consisten en el uso de los valores de activación de una de las capas ocultas de las DNNs como vectores de características. Los BF han demostrado funcionar muy bien en tareas LID (Song y col., 2013), (Matejka y col., 2014).
- Las *DNN posteriors* por otra parte, son una idea similar a la de los *GMMs posteriors* con la diferencia de que en lugar de generarse a partir de las estadísticas de un sistema típico de i-Vectores, estos se generan a partir de las probabilidades a posteriori de las clases modeladas por la DNN (Richardson, Reynolds y Dehak, 2015).

En lo que se refiere a las redes neuronales recurrentes se ha podido comprobar que su efecto recursivo es útil para el reconocimiento de idioma, dado que la información generada en el tiempo es capaz de combinarse con la información espectral, incorporando estados de memoria en la red y mejorando su rendimiento en la toma de decisiones. La razón es que incorpora información no solamente del estado presente sino de aquella generada en tiempos pasados. La base del funcionamiento de las RNNs está en el proceso de aprendizaje que utiliza el algoritmo de Back-Propagation Through Time (BPTT) (Werbos, 1990) capaz de procesar valores en tiempos precedentes al actual y por tanto, es capaz de introducir estados de memoria en nuestros sistemas.

Por estos elementos, creemos que adaptar las técnicas más exitosas en tareas de LID a las redes neuronales recurrentes nos puede ayudar a generar modelos innovadores que serán eficientes para tareas de identificación de idioma que utilizan secuencias de fonemas. Y que, al combinar las mejores técnicas, permitan obtener modelos capaces de proporcionar mejores tasas de reconocimiento.

1.2. Objetivos

La información de *contexto* en una frase es una característica propia de cada idioma que resulta de mucha utilidad para su comprensión y distinción. Si bien los elementos alofónicos que constituyen las frases de un idioma son fundamentales para su caracterización, con el contexto se posibilita su identificación. A lo largo de esta tesis se estudian parámetros y técnicas orientadas a utilizar de una manera más eficiente la información de contexto de las frases en las tareas de reconocimiento automático de idioma.

En lo que se refiere a parámetros, utilizamos unidades fonéticas que incluyen información de contexto (ngramas-fonéticos), así como coeficientes caracterizados por sus primeras y segundas derivadas que incluyen información de eventos pasados y futuros.

Respecto a las técnicas, usamos y adaptamos técnicas innovadoras que permiten incluir información de contexto en la tarea de reconocimiento. Hemos estudiado técnicas con las que podemos utilizar información del pasado para predecir eventos futuros en el desarrollo del habla (modelos de lenguaje basados en redes neuronales recurrentes), así como técnicas que permiten determinar la información de contexto más probable existente alrededor de un evento (embeddings neuronales).

Comprendemos por otra parte, que la definición de un idioma también viene dada por la frecuencia de ocurrencia de unos u otros sonidos. Por ello, utilizamos técnicas relacionadas con la categorización de textos (información discriminativa) para mejorar la tarea de reconocimiento.

Así, y para abarcar los ámbitos de estudio propuestos, utilizamos dos sistemas como base de experimentación, uno basado en i-Vectores y otro que se basa en una estructura PPRLM. La base de datos sobre la cual se evalúan las diferentes técnicas propuestas, es KALAKA-3, en la condición *plenty* y *closed* (Rodríguez-Fuentes y col., 2016).

Con el fin de organizar y precisar los ámbitos de estudio, a continuación se enumeran los objetivos:

1. Estudiar el efecto de incorporar información discriminativa de idioma en unidades llamadas posteriorgramas. La idea consiste en explorar métodos que permitan agregar información discriminativa, aquella que otorga más relevancia a las unidades fonéticas que aparecen con más frecuencia en un idioma y menos frecuencia en los restantes, en

unidades fonéticas denominadas posteriorgramas buscando mejorar con ello las condiciones de clasificación del reconocedor. Así mismo, se busca analizar el impacto de la información discriminativa en los posteriorgramas por medio del uso de un factor de suavizamiento. Por otro lado, se busca evaluar el impacto de eliminar o incluir información de poca ocurrencia en la tarea de reconocimiento. El estudio se desarrolla utilizando el sistema basado en i-Vectores. Los posteriorgramas mejorados se utilizan como vectores de características para la obtención de i-Vectores mediante el entrenamiento de la matriz de variabilidad total T . A continuación, los i-Vectores se usan como entradas de un clasificador Multi-Clase de Regresión Logística (MLR) mediante el cual se lleva a cabo la tarea de clasificación.

2. Estudiar los Phone Log-Likelihood Ratio Features (PLLRs) a nivel de estados fonéticos, el tratamiento de su contexto temporal y su optimización. La idea consiste en ampliar el contexto temporal en los PLLRs obtenidos a nivel de estado fonético por medio de la aplicación de una técnica similar a la SDC (Shifted Delta Cepstrum) pero aplicada a los PLLR, llegando a obtener con ello los SDPCs (Shifted Delta PLLR Coefficients). Lo que se busca con los SDPCs es relacionar un fonema actual con su respectivo pasado y futuro en aquellas tramas donde se posee la mayor cantidad de información espectral, de esta manera se logra modelar de mejor manera el contexto en una secuencia de fonemas y con ello favorecer la tarea de reconocimiento. El método utilizado para ello ha sido la optimización de los parámetros SDC buscando una configuración estándar que pudiera ser útil para diversas tareas.

3. Aplicar los modelos de lenguaje basados en redes neuronales recurrentes a nivel fonético en tareas de reconocimiento de idioma (RNNLM-P). Para ello, proponemos el uso de unidades fonéticas que incorporen en sí mismas información de contexto (ngramas fonéticos), lo que nos permite aumentar el inventario de unidades fonéticas y mejorar la tarea de reconocimiento. El sistema LID propuesto se basa en una estructura PPRLM, que cuenta con dos etapas, una de preprocesamiento, con la que se obtienen las secuencias fonéticas y, otra, donde a cada uno de los fonemas de la secuencia se le agrega información contextual creando un nuevo tipo de unidad fonética llamado ngrama-fonético, que, en conjunto, son los que se usan para el entrenamiento de los modelos de lenguaje y para la evaluación del sistema.

4. Adaptar la representación vectorial de palabras a un sistema fonotáctico para tareas de reconocimiento de idioma. La idea es utilizar la representación vectorial de unidades fonéticas (que para nuestro caso son los ngramas-fonéticos) para modelarla y usarla de manera similar a los i-Vectores, como vectores continuos y de baja dimensionalidad. Con ello podemos conseguir una caracterización más eficiente de dichas unidades.

Utilizaremos dos enfoques para conseguirlo, usaremos el modelado de las representaciones vectoriales de ngramas fonéticos, primero, como vectores de parámetros en tareas LID, y luego, como optimizadores de las secuencias utilizadas para el entrenamiento de RNNLM-P. Dentro del primer enfoque usaremos información de contexto local y de contexto global para el modelado. En tanto que en el segundo enfoque se usa la dispersión de información como elemento de análisis y la importancia de los ngramas-fonéticos en función de su aparición.

1.3. Organización

La tesis está organizada como sigue:

El Capítulo 2, Estado de la cuestión, se encuentra dividido en seis secciones. La primera se refiere a aspectos generales de la identificación de idioma, se describe la estructura PPRLM que nos ha servido como base para proponer un sistema en el cual los modelos de lenguaje se generen a partir de redes neuronales en vez del modelo de lenguaje clásico. Este sistema se describe en la segunda sección, principalmente, la fase feed-forward, la de retro-propagación y el proceso de factorización de la capa de salida. En la tercera sección, se describe la representación vectorial de unidades fonéticas y el modelo utilizado para transformar dichas representaciones en Embeddings Neuronales. En la cuarta sección se describe la estructura del sistema de reconocimiento basado en i-Vectores utilizado en esta tesis y las bases teóricas sobre las que se sustenta, entre ellas están: los supervectores, el JFA (Joint Factor Analysis), el SMM (Subspace Multinomial Model) y los propios i-Vectores. También en esta sección se describe el concepto de la categorización basada en n-gramas que resulta fundamental para comprender el concepto de información discriminativa, que incluimos en los sistemas fonotácticos de reconocimiento. La quinta sección está orientada a describir los coeficientes PLLRs y la técnica SDC, dado que son las ideas mediante las cuales se han creado los SDPCs. Con estos novedosos coeficientes se ha analizado el comportamiento del sistema de i-Vectores con diferentes variantes, tanto en la estructura como en la configuración de parámetros. Y por último, en la sexta sección se describen los procedimientos utilizados en esta tesis para calibrar y fusionar los scores de verosimilitud obtenidos de los sistemas de reconocimiento acústicos y fonotácticos con el fin de mejorar las tasas de reconocimiento.

En el capítulo 3 se describe la base de datos utilizada en esta tesis para evaluar las técnicas aplicadas. Hemos elegido una base de datos de tamaño y características compatibles con nuestra necesidad de entrenar en lapsos razonables de tiempo sistemas que cuentan con un elevado número de parámetros de configuración. También describimos la Función de Coste de Detección Promedio (C_{avg}) que es la métrica utilizada a lo largo

de la tesis para evaluar a los sistemas de reconocimiento propuestos. Por último, describimos el sistema acústico basado en MFCCs que hemos utilizado como línea base de comparación y mejora en base a los resultados obtenidos con las técnicas fonotácticas que se presentan en los capítulos posteriores.

En los capítulos 4, 5, 6 y 7 se presentan las líneas de investigación que hemos explorado y estudiado y que nos han permitido cumplir los objetivos propuestos en la presente tesis.

En el capítulo 4 se toman como referencia dos técnicas con las que cuenta nuestro grupo de investigación para el reconocimiento de idioma: la técnica de posteriorgramas y la de los rankings discriminativos de idioma. Con ellas, hemos buscado la manera de incorporar la información discriminativa de idioma en la generación de posteriorgramas. El sistema sobre el cual se realiza este trabajo es el basado en i-Vectores.

En el capítulo 5 partimos de los parámetros PLLR y les añadimos información de contexto temporal mediante la aplicación de la técnica SDC, formando nuevas unidades denominadas SDPCs. Así mismo, hemos propuesto configuraciones estables que se obtienen en la optimización de los parámetros SDC.

En el capítulo 6 se exponen los resultados de crear modelos de lenguaje basados en redes neuronales a partir de *ngramas-fonéticos* en tareas LID. De esta manera, se evalúan los efectos de la aplicación de estas unidades fonotácticas en RNNLMs determinando valores óptimos de configuración para *ngramas-fonéticos* de 1, 2 y 3 elementos, aprovechando la estructura de una red neuronal recurrente que utiliza el algoritmo de retropropagación en el tiempo como método de aprendizaje, el cual nos permite incorporar información de eventos pasados en la predicción de los siguientes.

En el capítulo 7 se realiza un estudio sobre la representación vectorial de unidades fonéticas, particularmente de los Embeddings Neuronales (ENs) resultantes de la aplicación de un modelado de representación vectorial (VSM) sobre *ngramas-fonéticos* y su uso en tareas de reconocimiento de idioma. En primer lugar, se evalúa el comportamiento de los sistemas de reconocimiento de idioma que utilizan como vectores de parámetros las secuencias de embeddings neuronales, y en segundo lugar, se estudian mecanismos para disminuir, por medio de los ENs, la dispersión de información provocada por el uso de *ngramas-fonéticos* de orden superior en la generación de modelos de lenguaje basados en RNNLMs.

En el capítulo 8 se exponen las conclusiones generadas a partir del trabajo desarrollado, así como las líneas futuras en el área de investigación.

Capítulo 2

ESTADO DE LA CUESTIÓN

2.1. La identificación de idioma y su clasificación

2.1.1. Generalidades

En general, una tarea LID puede describirse como un problema de clasificación multi-clase donde la meta es clasificar el lenguaje utilizado en una señal de voz, intentando emular el proceso de identificación de idioma realizado por los humanos. En un primer momento, el sistema *escucha* las señales de voz y a partir de ello determina el lenguaje al que se corresponden dichas señales. Si el idioma no le resulta familiar, entonces define parámetros de similitud que le aportan *pistas* del idioma con el que se corresponderían las expresiones (Muthusamy, Barnard y Cole, 1994).

La mayoría de los sistemas LID opera en dos fases: entrenamiento y evaluación. Durante la fase de entrenamiento, el sistema aprende las características de cada uno de los idiomas a reconocer en base a los vectores de características acústicos previamente etiquetados de forma supervisada; este aprendizaje le lleva al sistema a definir tantos modelos como idiomas a reconocer existan. Los parámetros, también llamados vectores de características que se utilizan para entrenar los modelos de idioma se obtienen aprovechando la cuasi-estacionalidad de las señales de audio en tramas de pocos milisegundos, de donde es posible extraer su espectro de frecuencia y a partir de él mediante un proceso de muestreo, cuantificación y filtrado obtener los elementos discretos característicos de la señal que la convierten en identificable. En (Cumani, 2012) se describe dicho proceso de manera bastante detallada. En general para los sistemas de reconocimiento de habla y específicamente en los sistemas LID se utilizan coeficientes Mel (Rabiner y Juang, 1993), (Vergin, O'shaughnessy y Farhat, 1999) ó PLPs (Hermansky, 1990) con este fin y más actualmente se usan coeficientes PLLRs (Diez y col., 2013). Así mismo y con el fin de extraer más información de estos coeficientes se han utilizado sus correspondientes valores Delta (derivadas), Delta-Delta (doble derivadas) (Bielefeld, 1994), (Ambikairajah

y col., 2011) que al ser utilizados en un contexto de más duración han mejorado considerablemente sus prestaciones como en (Torres-Carrasquillo y col., 2002) y en (D'Haro y col., 2014).

El habla se corresponde con un uso particular e individual que hace una persona de una lengua para comunicarse. Por lo tanto, es razonable considerar a la realización del habla en el tiempo como un proceso complejo definido por variables interdependientes que son características tanto del locutor como de las condiciones del canal de transmisión. Por dicha complejidad se considera al tratamiento del habla como un proceso estocástico estacionario a corto plazo dominado principalmente por la probabilidad de ocurrencia de un evento en un momento específico. Si se considera a los vectores de características como realizaciones de alguna variable aleatoria, se puede modelar la distribución presentada por éstos con un Modelo de Idioma que contenga las particularidades propias de cada uno y que sirva de base comparativa para las señales de voz utilizadas para evaluar el rendimiento del reconocedor (señales de evaluación).

Entre estos modelos existe una gran variedad que han sido categorizados de diferentes maneras en función de sus características. Singer (Singer y col., 2003) los clasifica como paramétricos o no paramétricos dependiendo si están basados en estructuras estáticas (plantillas) donde los datos de entrenamiento y de evaluación son directamente comparados unos con otros asumiendo que, uno es una réplica imperfecta del otro, o bien, en estructuras estocásticas donde cada lenguaje es modelado como una fuente probabilística de una ajustada pero desconocida función de densidad de probabilidad. En esta perspectiva, modelos como los obtenidos mediante Cuantificación Vectorial (VQ) o los obtenidos por medio de Dynamic Temporal Warping (DTW) son definidos como modelos no paramétricos, en tanto que modelos obtenidos a partir de Modelos de Mezcla de Gaussianas (GMMs) o de Modelos Ocultos de Markov (HMMs) se definen como paramétricos.

Más recientemente, Kinnunen (Kinnunen y Li, 2010) clasifica los modelos en función del objetivo a conseguir con el entrenamiento del modelo. Así, aquellos modelos que caracterizan las fronteras de decisión entre idiomas se denominan *discriminativos*, mientras que aquellos que estiman las distribuciones de los vectores de características de cada idioma se denominan *generativos*. Entre los modelos discriminativos aparecen aquellos obtenidos a partir de redes neuronales (NNs) y a partir de Support Vector Machine (SVMs), en tanto que como modelos generativos aparecen los obtenidos a partir de GMMs y de VQ.

Se utilizan uno o varios de los modelos aquí mencionados en la fase de evaluación de un sistema de reconocimiento de idioma. En la fase de evaluación, a una señal de voz de un idioma desconocido se le extrae su vector de características de la misma forma

FIGURA 2.1: Sistema LID básico

que en la fase de entrenamiento. Dicho vector se compara con cada uno de los modelos de idioma entrenados, escogiendo el idioma correspondiente al modelo de mayor verosimilitud. La estructura que ejemplifica el procedimiento descrito se presenta en la Figura 2.1.

Cabe mencionar aquí que el hecho de que la señal de evaluación pertenezca o no a uno de los idiomas a reconocer deriva en la aparición de dos tipos de evaluación de sistemas LID:

- Closed Set.- En esta evaluación se espera que solo los idiomas a reconocer aparezcan en las señales de evaluación.
- Open Set.- Las señales de audio de evaluación pueden ser de cualquier idioma, tanto del conjunto de idiomas a reconocer como de idiomas que estén fuera de dicho conjunto, con lo que el sistema debe dar una verosimilitud de que la señal no corresponda a ninguno de los idiomas a reconocer.

En función del tipo de información utilizada (Li, Ma y Lee, 2013), los sistemas de reconocimiento de idioma pueden ser de dos tipos: los sistemas acústicos, que utilizan información espectral para diferenciar entre idiomas (JFA (Kenny y col., 2008), i-Vectores (Dehak y col., 2011b), PLDA (Wang y col., 2009)), y sistemas fonotácticos (PPRLM (Zissman, 1996), RNNLMs (Mikolov y col., 2010), embeddings neuronales (Mikolov, Yih y Zweig, 2013)), DNNs (Lopez-Moreno y col., 2014), que modelan las probabilidades de las secuencias de fonemas obtenidas como salida de un conjunto de reconocedores de fonemas.

2.1.2. Sistemas LID acústicos

A los reconocedores que utilizan únicamente la información contenida en el espectro de frecuencia se les suele llamar reconocedores acústicos. Capturan las diferencias esenciales entre lenguajes por medio del modelado de las distribuciones de sus componentes espectrales. En general, este tipo de reconocedores suelen utilizar GMMs y SVMs para

representar los vectores de características de las señales de audio con las que se entrena o se evalúa el reconocedor. Para el caso de GMMs, en la fase de entrenamiento y para cada idioma se determinan sus respectivas distribuciones asumiendo que cada valor del vector de características se corresponde con una densidad de probabilidad y que el conjunto de vectores se corresponde con la suma ponderada de las densidades gaussianas multivariantes. En la fase de evaluación, se obtiene el vector de características de la señal de audio desconocida y se calcula la log-verosimilitud de cada uno de los modelos de lenguaje obtenidos en el entrenamiento.

En el caso de SVM, el modelo toma los vectores de características de entrada obtenidos a partir de las señales de voz y los mapea en un espacio de alta dimensionalidad donde se separan las clases por medio de un hiperplano (Campbell y col., 2006a). Los datos de los vectores de características mapeados que se encuentran en las fronteras de decisión son los *vectores de soporte*. El objetivo de SVM es precisamente modelar dichas fronteras. Dado que el SVM es un clasificador de dos clases (binario) se considera a las tareas LID como un problema de verificación, en donde se usa la estrategia de *uno contra todos los demás*. De esta forma, para entrenar un modelo de lenguaje con SVM, se inicia entrenando el modelo del idioma objetivo con los vectores de características de ese idioma definiéndolas como clase 1 y con los vectores de características de todos los otros idiomas a las que se les denomina clase 0. Estos datos son procesados por un SVM, usando un *kernel*, que se define como la métrica de similitud entre el modelo de una señal de audio 1 y una señal de audio 2. El resultado de todo el proceso es un modelo SVM que representa el idioma objetivo. El proceso se repite para los otros idiomas.

En (Lopez-Moreno y col., 2014) se evalúa el uso de DNNs a nivel acústico con gran éxito en su comparación con los sistemas clásicos basados en i-Vectores, en tanto que en (Matejka y col., 2014) se propone las *bottleneck features* (BF) que consisten en el uso de los valores de activación de una de las capas ocultas de las DNNs como vectores de características, consiguiendo con ellas mejorar la tarea de identificación de idioma.

2.1.3. Sistemas LID lingüísticos

Los sistemas LID que utilizan un reconocedor fonético, también llamados *lingüísticos*, se clasifican en función del nivel de información que utilizan para el reconocimiento (Figura 2.2). La información considerada de bajo nivel es aquella que utiliza elementos fonéticos de base (los fonemas) para el reconocimiento y todo tipo de información útil que se agregue a la información de base generan los niveles superiores. En un segundo nivel encontramos a los reconocedores fonotácticos, en los que además de los fonemas utilizados en los sistemas de bajo nivel utilizan información relacionada con la combinación de fonemas. Se valoran las secuencias fonéticas y la probabilidad de que

FIGURA 2.2: Niveles de Información usados por LID

éstas se correspondan más con un idioma que con otro. En un tercer nivel están los reconocedores prosódicos que estudian el acento particular de cada idioma. Están muy correlacionados con la frecuencia fundamental y consideran la duración, la energía y la entonación de los fonemas implicados. Los sistemas que se han mencionado hasta ahora, fonológicos, fonotácticos y prosódicos forman parte de un grupo de reconocedores que se pueden considerar como preléxicos dado que no utilizan elementos gramaticales para el reconocimiento. Aquellos que los utilizan se denominan léxicos, en los que es posible utilizar directamente las palabras o también la sintaxis de la frase para la tarea de reconocimiento. Con la información agregada se entrenan los modelos de lenguaje y se definen las características de cada uno de ellos, que son luego utilizadas por el clasificador para identificar el idioma.

En la presente tesis se realizará el estudio de sistemas LID con información de bajo nivel, principalmente con sistemas acústicos-fonológicos y fonotácticos partiendo de la base de los resultados obtenidos por (Moreno y col., 2014) al usar sistemas acústicos y de (D'Haro y col., 2013) con sistemas fonotácticos. La tarea LID, tanto para los reconocedores de tipo acústico-fonológico como para los de tipo fonotáctico, se llevará a cabo en dos fases: entrenamiento y evaluación. Por otro lado, se buscará adaptar los sistemas de reconocimiento de idioma de alto nivel a sistemas de bajo nivel con el fin de aportar información discriminativa de idioma al reconocedor.

Reconocedores fonéticos en paralelo seguidos de modelos de lenguaje (PPRLM)

PPRLM es un método popularmente utilizado en tareas de identificación de idioma (Zissman, 1996), (Cordoba y col., 2007). El objetivo de la aplicación de esta técnica es modelar la frecuencia de ocurrencia de las secuencias de fonemas en cada uno de los

FIGURA 2.3: Estructura de un sistema PPRLM para tareas LID

idiomas a reconocer. En esta tesis, la estructura PPRLM ha sido utilizada para evaluar el rendimiento de los modelos de lenguaje basados en redes neuronales.

Su estructura puede verse en la Figura 2.3 y se define en dos pasos:

- El primero consiste en la generación de N conjuntos de secuencias de fonemas mediante el uso del mismo número de reconocedores fonéticos que se ejecutan en paralelo para cada uno de los fragmentos de audio correspondientes a los idiomas que el reconocedor será capaz de identificar. Para el caso del ejemplo presentado en la Figura 2.3 se usan tres reconocedores de fonemas que funcionan en paralelo (húngaro, checo y ruso) y generan un conjunto de secuencias de fonemas para cada uno.

- El segundo consiste en el modelado de cada uno de los N conjuntos de secuencias de fonemas para cada uno de los idiomas. Concretamente, se asigna un *score* que refleja la probabilidad de que un conjunto de secuencias de fonemas analizado se corresponda o no con cada uno de los idiomas a reconocer. En el caso del ejemplo propuesto en la Figura 2.3 cada uno de los 3 conjuntos de secuencias fonéticas se modelan para los cuatro idiomas para los que está diseñado el sistema referido, español, portugués, italiano y francés, obteniendo con ello 12 modelos.

El idioma al que pertenece una señal de audio de entrada en la fase de evaluación L_{eval} lo define el máximo valor de verosimilitud encontrada entre la propia señal y cada uno de los modelos entrenados (D'Haro, 2009), así:

$$L_{eval} = \underset{s}{argmax} \sum P(u|s, L, \phi)P(s|L) \quad (2.1)$$

donde u es la señal de entrada de un idioma desconocido, s es la secuencia de fonemas, L es el conjunto de posibles idiomas a identificar y ϕ es el conjunto de modelos acústicos. $P(s|L)$ es la probabilidad a priori de la secuencia s dado el conjunto de lenguajes a identificar.

Una de las particularidades más significativas de la técnica PPRLM es que el conjunto de secuencias generadas por el reconocedor fonético de uno de los idiomas a reconocer no es necesariamente el conjunto que mejor modele dicho lenguaje, lo que convierte a PPRLM en una técnica versátil y adaptable a reconocedores fonéticos de diferentes idiomas diversificando su uso y aplicabilidad.

2.2. Modelos de lenguaje basados en redes neuronales recurrentes usando ngramas-fonéticos (RNNLM-P)

Los modelos de lenguaje fonotácticos se definen por medio del producto de probabilidades de ocurrencia de una sucesión de eventos fonéticos en una señal de voz. En este contexto, los sistemas basados en cuentas, en los que utilizando la frecuencia de ocurrencia de los eventos se determina la probabilidad de los mismos, se han venido utilizando con éxito en tareas de reconocimiento. Los métodos y técnicas para obtener dichas probabilidades han sido diversas, como las presentadas por Bell (Bell, Cleary y Witten, 1990) o las de Kneser (Ney, Essen y Kneser, 1994) o sus respectivas variantes (Kneser y Ney, 1995). Las mencionadas técnicas y otras de gran utilidad aún están vigentes y soportadas por varios tool-box de generación de modelos de lenguaje, como por ejemplo, el SRI-LM (Stolcke, 2002).

El uso de NNs en la generación de modelos de lenguaje la propuso Bengio en (Bengio y col., 2006) basándose en el principio de recursividad y de la representación del tiempo en redes neuronales propuesto por Jordan en (Jordan, 1986) y Elman en (Jeffrey, 1990) respectivamente. Los modelos de lenguaje basados en NNs se pueden generar gracias a las características de no-linealidad de la capa de estado de la red neuronal, que le permite actuar como detectores de patrones.

Con estos elementos, Bengio en (Bengio y col., 2006) propuso una representación distribuida para modelos de lenguaje usando redes neuronales artificiales de una capa de entrada y una sola capa de estado como la que se muestra en la Figura 2.4.

Dicha capa de estado posee características de respuesta no-lineal y contiene un conjunto de datos que actúan como *pesos* de cada una de sus neuronas. En esencia, en la capa de estado, cada neurona j puede ser vista como un filtro de correlación, el cual mapea la

FIGURA 2.4: Red neuronal de una capa de entrada y una de estado

entrada proveniente de todas las neuronas de la capa de entrada y activa una salida de tipo escalar y , de la forma:

$$y_j = \sigma(b_j + \sum_i \omega_{ij} z_i) \quad (2.2)$$

donde σ es una de las típicas funciones no-lineales, ya sea una función logística o una función de tangente hiperbólica y b_j es un bias. Fundamentalmente, la unidad j se dispara si la correlación entre las entradas y su peso excede un umbral $-b_j$. De esta forma, las neuronas de la capa de estado pueden ser vistas como detectores de patrones. Si a esto se le aplica una función softmax 2.3, los patrones detectados pueden ser considerados como probabilidades de ocurrencia. Tal como se puede ver en la Figura 2.3 donde la salida es la probabilidad de que se produzca el evento $w(t+1)$ dada la historia almacenada en la capa de estado $s(t)$.

$$y_j = \frac{\exp(y_i)}{\sum_k \exp(y_k)} \quad (2.3)$$

Al conseguirse que la respuesta de la red neuronal a un evento fonético sea una probabilidad condicional del siguiente evento a ocurrir, el producto de probabilidades conseguidas de una secuencia de eventos deriva en modelos de lenguaje. Aunque la idea resultaba bastante lógica no resultó del todo funcional, porque a pesar de que a la salida de la red se obtenía una probabilidad de ocurrencia del próximo evento a ocurrir, aún la NN no tenía la capacidad de recuperar información del pasado (Hanson y Burr, 1990) como para caracterizar adecuadamente las características de un idioma.

El uso de redes neuronales recurrentes (Mikolov y col., 2010), (Mulder, Bethard y Moens,

2015) que utilizan un método de aprendizaje que posibilita la recuperación de información del pasado para el modelado de lenguaje ha resultado clave para recuperar esa información contextual producida en cada instante en la capa de estado. Aprovechando el método de aprendizaje, se ha podido recuperar información de tiempos pasados dentro de la capa de estado de la red, con lo que teóricamente se tiene en cuenta información infinita relacionada con la historia. Si se retroalimenta esta información a la capa de entrada, la red introduce información de contexto para definir la probabilidad de aparición del siguiente evento.

La información obtenida del pasado se almacena en forma de representaciones vectoriales en la salida de la capa de estado en el tiempo $t-1$ y se colocan nuevamente en la capa de entrada (en una sección destinada para el efecto) y junto con las V entradas comunes de la red neuronal, forman la nueva entrada activada de la siguiente palabra en el tiempo t , la cual, después de ser procesada dentro de la red genera una distribución de probabilidad condicional que incorpora información del tiempo $t-1$ en su decisión.

Este proceso se puede ampliar a n tiempos anteriores con lo que la memoria de la red se incrementa tanto como se requiera. Entre los algoritmos más conocidos y utilizados están: el de Real-Time Recurrent Learning (RTRL) (Chang, Chang y Huang, 2002) y el de Back-Propagation Through Time (BPTT) (Werbos, 1990) , (Guo, 2013). Ambos utilizan el descenso del gradiente como método de aproximación del error a cero, condición necesaria para ajustar y optimizar los pesos con los que la red se entrena y que son los que le permiten tener la capacidad de reconocer vectores de entrada de idioma desconocido (Hochreiter, 1998).

Los modelos utilizados para las tareas de reconocimiento de idioma deben ser compatibles con las características de lenguaje natural (Manning y Schütze, 1999) y en ese contexto, la información recuperada del pasado resulta clave. La recursividad y los algoritmos de aprendizaje capaces de recuperar esta información, potenciaron el uso de redes neuronales para el reconocimiento de voz y de idioma. Sin embargo, respecto a los algoritmos utilizados en la fase de aprendizaje existe el problema del rápido desvanecimiento del gradiente (Bengio, Simard y Frasconi, 1994) ó (Hochreiter y col., 2001) y por ello, se han planteado diversas alternativas, teniendo entre las más importantes al *Long Short Term Memory* (LSTM) (Sundermeyer, Schlüter y Ney, 2012), a la regularización (Zaremba, Sutskever y Vinyals, 2014) ó al aprendizaje de memoria extendida (Mikolov y col., 2014). Así, a fecha de hoy las redes neuronales recurrentes constituyen una de las principales herramientas para la generación de modelos de lenguaje (Mikolov y col., 2010), (Mulder, Bethard y Moens, 2015) y se perfila como un modelo muy útil para tareas de identificación de idioma (Liu y col., 2014) ó (Gonzalez-Dominguez y col., 2014).

En general, los modelos de lenguaje basados en redes neuronales recurrentes (RNNLMs)

FIGURA 2.5: Generación de los ngramas-fonéticos a partir de una secuencia de fonemas

están diseñados para modelar eventos a nivel léxico y en ese escenario resultan claramente eficientes e.g.(Mikolov y col., 2010). Sin embargo, al tratar con unidades más pequeñas como el caso de fonemas en el plano fonotáctico, se requiere el uso de varias capas para modelar adecuadamente la historia pasada y los distintos contextos en los que pueden aparecer dichas unidades (Hwang y Sung, 2016), por lo que se requiere de un mayor número de datos para conseguir un buen entrenamiento. Todo esto incrementa el costo computacional, además de que el sistema se expone a un sobre-entrenamiento por las características que adquiere el modelo (Zaremba, Sutskever y Vinyals, 2014).

Para adaptar los RNNLMs a un nivel fonético-fonotáctico proponemos el uso de vectores de características que aglutinen en una sola unidad a dos o más fonemas. La idea nace de utilizar parámetros que implícitamente posean información contextual para entrenar los modelos de lenguaje y que a la postre favorezca el reconocimiento de idioma. A estos modelos los hemos denominado RNNLM-P, entendiendo que los modelos se generan a partir de parámetros que incluyen información fonotáctica. Para ello, utilizamos unidades fonéticas llamadas *ngramas-fonéticos* que aglutinan en una sola unidad a dos o más fonemas de una secuencia de fonemas. En la Figura 2.5, se puede ver cómo la secuencia de fonemas $u\ n:\ i\ J\ o\ n$ se convierte en la secuencia de 2gram $u_n:\ n:_i\ i_J\ J_o\ o_n$. ó, en la secuencia de 3gram $u_n:_i\ n:_i_J\ i_J_o\ J_o_n$. Si tenemos en cuenta la primera unidad fonética para los tres casos, u para fonemas, $u_n:$ para 2gram y $u_n:_i$ para 3gram, podemos ver cómo se incluye implícitamente la información de contexto de idioma en cada una de ellas.

El hecho de *agrupar* fonemas adyacentes en una nueva unidad fonética conlleva la aparición de nuevas unidades, generando un incremento del inventario de unidades fonéticas. Así, por ejemplo, para la base de datos KALAKA-3 utilizada en esta tesis, (Rodríguez-Fuentes y col., 2016) y usando el reconocedor de fonemas de la Universidad de Brno (Schwarz, 2009), los ngramas-fonéticos de 1 elemento generaron un vocabulario de 61 elementos (igual al del modelo original), los ngramas-fonéticos de 2 elementos

generaron uno de 1938 elementos y los de 3 elementos uno de 28097 elementos. Todos ellos obtenidos en el entrenamiento de datos de Español con el modelo del reconocedor fonético de húngaro.

2.3. El modelado en el espacio vectorial

2.3.1. Descripción

Los modelos en el espacio vectorial (embeddings neuronales) representan a las unidades fonéticas de un corpus en el espacio continuo de baja dimensión, donde dichas unidades similares en el ámbito semántico se proyectan de una manera particular dentro de este espacio. Se modelan diagonalizando la matriz que las contiene. Así, entendiendo en primera instancia que para nosotros las unidades fonéticas equivalen a palabras en el modelo, las palabras que aparezcan en los mismos contextos compartirán significado semántico.

Los embeddings neuronales han recibido interés particular en los sistemas de procesamiento de lenguaje natural debido a las relaciones semánticas y sintácticas que se llegan a encontrar entre palabras. Tradicionalmente en los sistemas mencionados, se trata a las palabras como símbolos. Así por ejemplo, la palabra *rey* puede ser representado por Id001, en tanto que *reina* por Id234. Esta codificación es arbitraria y provoca que en el espacio vectorial, una y otra queden en cualquier lado, sin ningún tipo de orientación. En cambio, una representación vectorial de estas palabras, al entrenarse con valores reales y no codificados puede ayudar a crear un tipo de relación en el espacio vectorial con la cual se pueden entrenar modelos estocásticos. Así, la representación vectorial de palabras se ha utilizado en diferentes áreas relacionadas con la tecnología de habla, como por ejemplo, en *language understanding* (Yao y col., 2013) ó en *spoken language translation* (Le y col., 2016).

Su aplicación se ha estudiado desde varios enfoques considerado principalmente dos categorías: los métodos basados en cuentas, como por ejemplo LSA (Latent Semantic Analysis) (Wiemer-Hastings, Wiemer-Hastings y Graesser, 2004) que otorga contexto semántico a palabras, agrupándolas según su significado y posibilitando la reducción de dimensionalidad, y por otra parte, los métodos predictivos, aquellos que principalmente vienen generados a partir de redes neuronales y de modelos de lenguaje, en los que se busca predecir el estado del siguiente evento a ocurrir en base a los eventos pasados, como por ejemplo, los modelos propuestos por Bengio (Bengio y col., 2006) ó Mikolov (Mikolov y col., 2013c).

Entre las técnicas más comunes que se utilizan para la generación de embeddings neuronales (modelado de las representaciones vectoriales) tenemos: los modelos Skip-Gram y C-Bow, que toman en cuenta el contexto local definido por los elementos cercanos al analizado y los modelos GloVe (Global Vectors) (Pennington, Socher y Manning, 2014), LSA (Wiemer-Hastings, Wiemer-Hastings y Graesser, 2004), Hellinger PCA (Principal Component Analysis) (Lebret y Collobert, 2013) ó embeddings basados en PPMI (positive pointwise mutual information) (Levy y Goldberg, 2014) que tienen en cuenta contextos globales mediante las cuentas realizadas sobre los elementos existentes en el corpus.

El modelo GloVe busca reunir las características de aquellos métodos que utilizan la matriz de factorización (ventanas de contexto alrededor del corpus de datos) para establecer relaciones semánticas y sintácticas entre palabras junto con las características de métodos más específicos que utilizan la analogía de palabras (mediante el uso de ventanas de contexto locales) en un único método que utilice lo mejor de los dos. Para ello, Pennington (Pennington, Socher y Manning, 2014) propone modelos de mínimos cuadrados ponderados que entrenan las cuentas de co-ocurrencia palabra-palabra, haciendo un uso eficiente de las estadísticas del corpus de datos.

El modelado de las representaciones vectoriales a través de redes neuronales se consigue usando métodos de entrenamiento basados en modelos de lenguaje como los propuestos por Bengio en (Bengio y col., 2006) y extendido por (Turian, Ratnoff y Bengio, 2010) o Mikolov (Mikolov y col., 2010) junto con la aplicación de las técnicas Skip-Gram y C-Bow. El objetivo de los ENs en este caso es predecir la siguiente unidad fonética a aparecer en función del contexto que incluye la unidad fonética de entrada.

- El Modelo Skip-Gram (Guthrie y col., 2006) elige aleatoriamente uno de los elementos que rodean al elemento de entrada analizado (elementos a los que denominamos *contexto*) y utiliza su embedding como representación del contexto.

- El modelo CBOW (Continuous Bag of Words) (Wu, Hoi y Yu, 2010) que usa el embedding promedio de los elementos del contexto del elemento de entrada analizado como representación de contexto.

Teóricamente, la distancia entre palabras que tienen una sintaxis y una semántica similar tiende a ser pequeña, en tanto que la distancia entre palabras de sintaxis y semántica diferente tiende a ser mayor. En base a este concepto se han desarrollado varias aplicaciones, tales como: la recuperación de la información de documentos a partir de palabras clave (Mogotsi, 2010), la clasificación de documentos en función de categorías predeterminadas (Sebastiani, 2002), sistemas de pregunta-respuesta (Tellex y col., 2003), entre otros.

Sin embargo, dado que las características semánticas y/o sintácticas no se encuentran

a nivel fonotáctico o fonético, lo que proponemos dentro de esta tesis y en este ámbito en particular es la adaptación de los modelos de representación vectorial a un nivel fonético, buscando que el cálculo de los embeddings se realice por una parte, en base a la co-ocurrencia de *ngramas-fonéticos* (que son nuestras unidades de estudio) y sus vecinos a lo largo del corpus de entrenamiento (Lebret, 2016) utilizando los modelos GloVe y, por otra parte, en base a predicciones de eventos que tienen en cuenta la información de contexto local (*ngramas-fonéticos* encontrados alrededor del *ngrama-fonético* analizado) para luego proyectar las cuentas obtenidas en vectores de baja dimensión para cada *ngrama-fonético* existente en el vocabulario.

Respecto al modelado por medio de ventanas de contexto locales, cuando los modelos de lenguaje se generan mediante redes neuronales, el problema de la complejidad computacional provocada principalmente por la función de activación no lineal en la capa de estado de la red se aborda aplicando técnicas de modelado, como Skip-Gram (Guthrie y col., 2006) y C-Bow (Wu, Hoi y Yu, 2010), las cuales, a pesar de formar parte de una estructura de red neuronal, no requieren de una función de activación no lineal en la capa de estado (Soutner y Müller, 2014) y así la complejidad computacional se ve disminuida (Turian, Ratinov y Bengio, 2010). De entre las dos, analizaremos en profundidad la técnica Skip-Gram que es la que mejores resultados ha generado en las pruebas preliminares que hemos llevado a cabo ya en el ámbito de la identificación de idioma.

2.3.2. Modelo Skip-Gram

El modelo Skip-Gram es una red neuronal clásica, cuya función de activación no lineal ha sido eliminada y donde la normalización Soft-max (Mikolov y col., 2013c) se ha reemplazado por una Soft-max de tipo jerárquico (Morin y Bengio, 2005).

El objetivo de entrenamiento de un modelo Skip-Gram es predecir, dentro de la misma frase, los *ngramas-fonéticos* de contexto que rodean al *ngrama-fonético* de entrada analizado. A diferencia del modelo de Bengio (Bengio y col., 2006), donde se predice la probabilidad del siguiente evento a ocurrir en base de la información almacenada en el vector de contexto que es el vector que posee la información histórica que se concatena con el vector de entrada de la red, al aplicar la técnica Skip-Gram se ignora dicho vector y se fuerza al modelo a predecir *ngramas-fonéticos* muestreados aleatoriamente desde la información contenida en un contexto de tamaño de ventana v (Le y Mikolov, 2014).

La representación vectorial de cada *ngrama-fonético* se entrena para maximizar la probabilidad logarítmica de los *ngramas-fonéticos* vecinos *de contexto* en una frase (Kusner y col., 2015) ó (Soutner y Müller, 2014), lo que creemos que puede favorecer el reconocimiento de idioma. En la Figura 2.6 se muestra este concepto.

FIGURA 2.6: Modelo Skip-Gram

Así, dado un conjunto T de ngramas-fonéticos ω y su contexto c (el contexto en el conjunto T es representado por uno de los ngramas en la ventana v donde se incluye el ngrama ω) se considera el cálculo de la probabilidad condicional $p(c|\omega)$. En el proceso de entrenamiento se definen los parámetros θ de $p(c|\omega; \theta)$ que maximizan la probabilidad:

$$\arg \max \prod_{\omega \in T} \prod_{c \in C(\omega)} p(c|\omega; \theta) \quad (2.4)$$

donde $C(\omega)$ es el conjunto de ngramas vecinos del ngrama ω y $p(c|\omega; \theta)$ es la probabilidad condicional asociada a la representación vectorial del ngrama ω y la representación de su contexto c .

Partiendo de la información del ngrama-fonético analizado y de la representación de su contexto dentro de la ventana v , al aplicar Skip-Gram se consigue predecir la probabilidad que maximiza el producto escalar entre vectores de ocurrencia frecuente que están dentro del contexto del ngrama-fonético de entrada y minimiza la probabilidad de ocurrencia de vectores poco frecuentes dentro de ese mismo contexto (Levy y Goldberg, 2014).

2.3.3. Negative Sampling (NS)

A partir de la definición del modelo para la generación de embeddings neuronales se han estudiado métodos de optimización del modelo. Entre ellos, podemos mencionar, el soft-max jerárquico (Morin y Bengio, 2005), la matriz de factorización (Levy y Goldberg, 2014) y el muestreo negativo (Goldberg y Levy, 2014). De entre ellos, la técnica de *negative sampling* (NS) nos ha parecido la más compatible con el Skip-Gram, dado que busca,

por un lado, que el valor de probabilidad calculado para el modelo de lenguaje maximice su producto escalar (distancia coseno) entre el embedding evaluado y su contexto cuando se haya visto dicho embedding en la etapa de entrenamiento, y que por otro lado, minimice el producto escalar entre el embedding evaluado y un contexto formado por embeddings aleatorios cuando el embedding no se ha visto en el entrenamiento. Este enfoque de optimización ha llevado a algunos autores a definir la técnica directamente como Skip-Gram-Negative-Sampling (SGNS) (Mikolov y col., 2013b).

El NS busca mejorar la robustez del embedding neuronal estimado mediante la aplicación de regresión logística, buscando resaltar la señal de voz sobre una señal de ruido generada con k ngramas escogidos aleatoriamente de la sentencia evaluada.

2.3.4. Modelos GloVe

GloVe pertenece a la familia de modelos que se usan para aprender vectores de palabras donde se captura las estadísticas del corpus global de forma directa. El modelo GloVe normalmente es utilizado a nivel de palabra, pero en nuestro caso lo vamos a evaluar a nivel fonético usando ngramas-fonéticos. En general, es un modelo similar al Skip-Gram con el elemento adicional de que utiliza ventanas de contexto globales utilizando la co-ocurrencia de elementos, con lo que captura directamente las estadísticas globales del corpus. La matriz de cuentas de co-ocurrencia de fonema-fonema es llamada X y las entradas X_{ij} tabularán el número de veces que el fonema j ocurrirá en el contexto del fonema i .

Por otro lado, $X_i = \sum_k X_{ik}$ es el número de veces que cualquier fonema aparece en el contexto del fonema i . Y siendo $P_{ij} = P(j|i) = X_{ij}/X_i$ la probabilidad de que el fonema j aparezca en el contexto del fonema i .

Considerando dos fonemas i y j que posean cada uno un interés particular, la relación entre ellos podrá ser examinada estudiando la tasa de sus probabilidades de co-ocurrencia con varios fonemas de prueba k . Con esta tasa se puede distinguir entre fonemas relevantes y no relevantes, así como para discriminar entre ellos (Pennington, Socher y Manning, 2014).

Así, considerando que la tasa P_{ik}/P_{jk} depende de los tres fonemas i, j y k , el modelo en su forma más general queda definido tal como se describe en la ecuación 2.5.

$$F(\omega_i, \omega_j, \omega_k) = P_{ik}/P_{jk} \quad (2.5)$$

Donde ω_i y ω_j son vectores de fonemas, en tanto que ω_k son vectores de fonemas de contexto.

Para operar la función F , se aprovecha que los espacios vectoriales son estructuras lineales para aplicar un producto escalar en la parte izquierda de la ecuación 2.5 y así tratar a F como valor escalar. Por otro lado, en las matrices de co-ocurrencia fonema-fonema, la distinción de fonemas y fonemas de contexto es arbitraria con lo que sus roles pueden ser intercambiables, en este proceso se pierde la simetría del modelo que luego se recupera por medio de valores de bias, b_i para ω_i y b_j para ω_j . La principal desventaja que presentaba este modelo en un inicio era que ponderaba a todas las co-ocurrencias por igual, incluso aquellas que ocurrían rara vez o nunca. Por ello, Pennington propuso un modelo de regresión de mínimos cuadrados ponderados introduciendo una función de pesos $f(X_{ij})$ en la función de costo final, con la que corregía dicho problema. La ecuación que define al modelo GloVe se expone a continuación en la ecuación 2.6.

$$J = \sum_{i,j=1}^V f(X_{ij})(\omega_i^T \omega_j + b_i + b_j - \log X_{ij})^2 \quad (2.6)$$

Donde T representa la transpuesta. Para mayores detalles se puede consultar el trabajo desarrollado por Pennington (Pennington, Socher y Manning, 2014).

2.4. Los i-Vectores y la información discriminativa entre idiomas como parámetros en tareas LID

2.4.1. Supervectores

En tareas de reconocimiento de idioma es importante conocer las mejores formas para representar señales de voz, que en general, tienen un número variable de parámetros. En este contexto, en la actualidad se viene utilizando el concepto de supervector, que constituye una forma robusta de presentar un conjunto de distribuciones utilizando un solo vector. Se define como un vector de características de dimensión alta y de tamaño fijo (independiente del tamaño de la señal de entrada) que permite modelar la distribución de probabilidad de parámetros extraídos a lo largo de un segmento/fichero de audio (Reynolds, 2006). Es un vector de características que concatena los vectores de medias de un UBM.

En general, es común crearlos a partir de los estadísticos de orden 0 o 1 de los coeficientes cepstrales y sus derivadas usando las mezclas de gaussianas entrenadas (Kinnunen y Li, 2010). Se les llama supervectores por el tamaño de los vectores de características resultantes, por ejemplo, en una configuración típica con 56 parámetros cepstrales y 512 gaussianas, su tamaño es de 28k.

Los supervectores, que en esencia representan los vectores de parámetros de las fuentes de información que alimentan los sistemas LID pueden ser tratados de dos formas, como si se tratara de un fenómeno continuo (Kenny y col., 2008) o como si se tratara de un fenómeno discreto (Soufifar, 2014). La diferencia esencial entre uno y otro radica en el tipo de distribución a utilizar para representar la información. Para el primer caso la distribución es multigaussiana, mientras que para el segundo la distribución es multinomial, asumiendo un espacio multi-dimensional para ambos casos.

Se ha vuelto una práctica casi común en tareas de reconocimiento la generación de supervectores para el tratamiento de señales (Dominguez, 2011).

2.4.2. Joint Factor Analysis

El *apilamiento* de información diversa en un supervector generada por la combinación de vectores de diferente procedencia, provoca que el efecto de canal aumente, apareciendo componentes espectrales debidos a cambios, tanto en las condiciones de grabación (e.g. micrófonos distintos o distancia variable entre el locutor y el micrófono), del locutor (e.g. cambios en la voz), o de ambiente (e.g. ruidos) que no se corresponden con la señal de voz que se tiene interés en procesar. Es de mucho interés para el procesamiento de habla y en este caso para la identificación de idioma, clasificar dichos sonidos y separarlos de la señal de voz. La separación de sonidos que no se corresponden con la señal de voz se controla con la denominada variabilidad de sesión.

Se entiende como variabilidad de sesión a todo fenómeno que provoque que dos grabaciones de un locutor en particular suenen de forma diferente. Normalmente este fenómeno es atribuible a efectos del canal de transmisión, por lo que en adelante a la variabilidad de sesión se le considerará como variabilidad de canal. El modelo que permite caracterizar la variabilidad de locutor y de canal como la combinación lineal de locutor/idioma y de canal se llama *Joint Factor Analysis* (JFA).

El modelo JFA utiliza modelos GMM que se han generado a partir del enfoque UBM-GMM propuesto por (Reynolds, Quatieri y Dunn, 2000).

Sin embargo, aparecieron problemas con UBM-GMM y MAP cuando al final se tenía que recurrir a técnicas no lineales para compensar las diferencias en el canal (e.g. z-score), así como cuando se tenían pocos datos para determinados idiomas y las variaciones entre los modelos resultantes no eran significativas, resultaba muy difícil separarlos. Con nuevas técnicas y especialmente con JFA se aprovecha de mejor manera el conjunto de datos a nivel de tramas y por tanto, se diferencian mejor los modelos resultantes. Así, se empezaron a utilizar combinaciones lineales de las direcciones más relevantes extraídas del análisis de la varianza del conjunto de datos de entrenamiento, con lo que se

introdujo el enfoque de *eigenvoice MAP* (Lucey y Chen, 2003), (Kenny, Boulianne y Dumouchel, 2005) y *eigenchannel MAP* (Kenny, Mihoubi y Dumouchel, 2003) para definir la señal de voz y la señal de canal respectivamente, dentro de un supervector. Fue esto último, lo que abrió el paso a la representación en un espacio de baja dimensionalidad de un conjunto de datos de alta dimensionalidad.

Joint Factor Analysis es un modelo matemático continuo que busca separar mediante el uso de modelos multi-dimensionales los efectos de la variabilidad de locutor/idioma y de sesión en Modelos de Mezclas de Gaussianas (GMMs).

El modelo JFA asume tres características que condicionan su aplicabilidad:

1. Un supervector M dependiente de locutor/idioma y canal, que se obtiene concatenando los vectores de media n -dimensional en el GMM correspondiente a una señal de audio dada, puede ser descompuesto por la suma de dos supervectores, un supervector de locutor S y un supervector de canal C ;

$$M = S + C \quad (2.7)$$

donde S y C son estadísticamente independientes y de distribución normal.

2. La distribución de S también denominada modelo de variabilidad de locutor/idioma, se describe de la siguiente manera:

$$S = m + T\omega + Dz \quad (2.8)$$

donde m es un supervector independiente del locutor/idioma y del canal (normalmente se obtiene a partir de un UBM). T es una matriz de dimensión alta que representa al subespacio de variabilidad del locutor/idioma en dimensión alta, ω son los Factores de Locutor/Idioma, que representan la variabilidad del locutor/idioma en baja dimensión y D es una matriz de dimensionalidad alta que junto a z , representa el *offset* de la media u , resultado de la adaptación MAP realizada sobre los UBM-GMMs y definida como el término residual de locutor/idioma.

3. La distribución de C que se corresponde con la componente dependiente de sesión/canal en forma de supervector de media, se describe de la siguiente manera:

$$C = Ux \quad (2.9)$$

donde u es una matriz que juega un papel similar a T , esta vez representando al subespacio de variabilidad de sesión. La variable x análoga a ω representa a los factores de canal.

Con lo anterior, el supervector dependiente de locutor/idioma y canal modelado con JFA se define así:

$$M = m + T\omega + Dz + Ux \quad (2.10)$$

2.4.3. I-Vectores

Es posible juntar las ventajas de los *eigenvoices* MAP y de la adaptación a posteriori de los UBM-GMMs tal como lo describe (Kenny y col., 2008) con el fin de mejorar el rendimiento del modelo. Considerando, sin embargo, las siguientes condiciones:

- Que los *eigenvoices* MAP resultan útiles cuando se tienen disponibles pocos datos de entrenamiento, que es una de las condiciones clásicas cuando se realizan tareas LID, y
- Que al adaptar a posteriori los UBM-GMMs se benefician únicamente aquellas Gaussianas que han sido observadas en los datos de entrenamiento, situación que resulta desfavorable en tareas de LID, dado que en éstas es habitual que vectores de entrada de uno u otro idioma no hayan aparecido en el entrenamiento.

Se considera conveniente eliminar el factor Dz de la expresión 2.10 (Kenny y col., 2008) con lo que el supervector de entrada quedaría como:

$$M = m + T\omega + Ux \quad (2.11)$$

Por otra parte, si se integran los efectos de variabilidad de canal en sus correspondientes de locutor en la matriz T , la ecuación 2.11 queda definida de la siguiente manera:

$$M = m + T\omega \quad (2.12)$$

donde m es el supervector independiente de locutor/idioma y de canal, que para nuestro caso es el supervector de medias de un UBM previamente entrenado, T es una matriz rectangular que contiene los eigenvectores con los *eigenvoices* MAP más grandes de la matriz de covarianzas de variabilidad total y proyecta un subespacio de baja dimensión, y por último ω que es una variable oculta que está definida por la distribución condicional posterior a las estadísticas Baum-Welch de una señal de voz dada. Esta distribución posterior es una distribución gaussiana y la media de ésta se corresponde con lo que es un i-Vector. El i-Vector es un vector que tiene una distribución normal estándar $N(0, I)$; y cuyos componentes son los factores que mejor describen el desplazamiento de media dependiente del vector de entrada (San-Segundo y col., 2016).

Cada componente del vector ω es tratado como una variable oculta donde la estimación de su MAP deriva en la idea del i-Vector. T puede ser estimado con los vectores ω buscando maximizar la verosimilitud de los datos de entrenamiento. Después de entrenar el subespacio, T se utiliza para extraer los i-Vectores ω de las señales de evaluación

(Dehak y col., 2011b). T proyecta un subespacio lineal en el espacio del supervector original y ω es la variable oculta de baja dimensión. El i-Vector es la estimación puntual de ω obtenido al adaptar el modelo de un vector de entrada dado. La ecuación 2.12 representa una versión simplificada de JFA que introduce el concepto de la variabilidad total, que consiste en un único subespacio en lugar de los dos propuestos en el JFA clásico. En él, se contienen simultáneamente las variabilidades de canal y de locutor/idioma sin distinción de los efectos producidos por uno u por otro, justificando esta hipótesis en el hecho de que los efectos producidos por el canal podrían contener también información de locutor/idioma.

La técnica de i-Vectores puede ser vista como la aplicación de la versión simplificada de JFA que permite proyectar un segmento de voz que normalmente está expresado por un supervector en un espacio de variabilidad de locutor/idioma y canal de dimensión alta en un vector en un espacio de variabilidad total de dimensión baja.

En (Dehak y col., 2011a) se demostró que la técnica JFA no separaba adecuadamente la información de locutor y sesión ya que encontraron que con el modelo de sesión era posible identificar al locutor con una alta tasa de acierto. Lo anterior hizo pensar que al aplicar JFA se perdía cierta cantidad de información discriminativa del locutor en el espacio del canal. Por tal razón y como se describe en (Kanagasundaram y col., 2014) se estudiaron técnicas de compensación de canal, entre las que cabe mencionar LDA (Linear Discriminant Analysis), WCCN (Within-class covariance normalization) y NAP (Nuisance attribute projection) para atenuar la variabilidad de canal en el espacio de i-Vectores.

2.4.4. Modelado en el Subespacio Multinomial

Como se mencionó en los apartados previos, los supervectores también se pueden tratar como un fenómeno discreto, entendiendo que la distribución que se les aplica es multinomial. En este contexto, aparece el modelado en el subespacio multinomial (SMM) (Kockmann y Ferrer, 2010). En general, para tareas de LID cada segmento de voz de entrada se puede representar mediante un vector que contenga las correspondientes estadísticas discretas de n-gramas. La verosimilitud logarítmica de un conjunto de segmentos de voz representado por vectores de entrada estará dada por la ecuación 2.13:

$$\sum_{1 \leq n \leq N} P(x_n | \lambda_n) \quad (2.13)$$

donde N es el número total de eventos (e.g. n-gramas que aparecen en la frase), x_n es el vector de cuentas de n-gramas multi-dimensional, λ_n es el parámetro del modelo

asociado al vector de entrada y $P(x_n|\lambda_n)$ es la probabilidad de ocurrencia de un determinado evento en el instante n considerando el modelo λ .

La verosimilitud logarítmica del segmento de voz k , representada por un vector de entrada de n -gramas C -dimensional, puede ser calculada como se muestra en la ecuación 2.14:

$$\log P(x_n|\lambda_n) = \sum_{1 \leq c \leq C} x_{nc} \log \lambda_{nc} \quad (2.14)$$

Donde x_{nc} es el valor de cuenta c del n -grama ω en el segmento k . λ_{nc} es el parámetro del modelo dependiente del vector de entrada que representa la probabilidad del correspondiente n -grama, y puede ser calculado por medio del modelado subespacial de las cuentas de n -gramas tal como muestra la ecuación 2.15:

$$\lambda_{nc} = \frac{\exp(m_c + t_c \omega_c)}{\sum_{1 \leq i \leq C} \exp(m_i + t_i \omega_n)} \quad (2.15)$$

donde w_n es una variable latente dependiente del vector de entrada cuya adaptación a posteriori genera los i-Vectores y t_c es la c fila de la matriz subespacial T que proyecta un subespacio lineal de baja dimensión en el dominio de la probabilidad logarítmica. Dados los parámetros m y T se puede estimar los valores necesarios de ω para maximizar la verosimilitud logarítmica en 2.14. Al igual que en el caso continuo, el modelo en el subespacio multinomial es usado como extractor de parámetros y cada i-Vector puede ser visto como una representación en baja dimensión de todo el vector de entrada. Los parámetros del modelo son estimados usando el método de máxima verosimilitud (ML). Inicialmente, los parámetros m y T necesitan ser estimados desde los datos de entrenamiento y para ello se sigue un proceso iterativo donde se alterna estimando ω con un extractor fijo T y luego estimando T con ω fijos. El valor de m se mantiene fijo durante la estimación.

El modelo se inicializa estimando por una parte las distribuciones multinomiales para los GMMs individuales utilizando todos los datos de entrenamiento, con lo que esto se traduce en sumar todos los supervectores de entrenamiento y luego normalizarlos sobre los rangos correspondientes a los GMMs individuales. Los supervectores de funciones multinomiales se denominan svUBM y el vector m se inicializa con el valor del logaritmo de svUBM. Todos los vectores ω son inicializados a cero y T es inicializada con la matriz de covarianza de los eigenvectores calculada de los supervectores de expresión suavizada svUBM centrado alrededor del vector m .

2.4.5. Categorización basada en n-gramas

En tareas de LID, la discriminabilidad es un factor que puede mejorar el rendimiento de los sistemas de reconocimiento, (BenZeghiba, Gauvain y Lamel, 2009), (Salamea-Palacios y col., 2013). Para lograrlo, se parte de la idea de que si puede hacerse más positivo el *score* para una hipótesis correcta y menos positivo para una hipótesis falsa, el reconocedor será más robusto y preciso. En (Cavnar y Trenkle, 1994) se propuso una técnica supervisada para tratar los diversos tipos de errores textuales que aparecen en los sistemas de categorización de texto. Esta técnica resultó ser muy efectiva, simple y robusta. Se basa en el cálculo y la comparación de la frecuencia de aparición de n-gramas en un texto.

La técnica se aplica en dos momentos (entrenamiento y evaluación). En la fase de entrenamiento, se leen los textos de entrenamiento por idioma y paralelamente se construye una plantilla de dos elementos, el listado de los n-gramas existentes en los textos y la frecuencia de ocurrencia o ranking *de cada uno de ellos*. Las plantillas se van completando a medida que continúa la lectura de los textos de entrenamiento, quedando en los primeros puestos del ranking aquellos n-gramas que han aparecido con más frecuencia en los textos del idioma a modelar pero menos en el resto de idiomas y en los últimos puestos aquellos que han ocurrido con menos frecuencia en el idioma a modelar. Estas plantillas pasan a ser modelos de los textos de entrenamiento y al tener uno por idioma pueden considerarse como modelos de lenguaje. En la etapa de evaluación, los textos se analizan de forma similar y se crea una plantilla de evaluación que tiene el listado de n-gramas y su frecuencia de ocurrencia (Cavnar y Trenkle, 1994).

2.5. Parámetros PLLR y Coeficientes Cepstrales con información temporal

2.5.1. Parámetros PLLR

En (Díez y col., 2012) se propone el uso de las probabilidades a posteriori obtenidas a la salida de decodificadores fonéticos *phone posterior* para ser utilizadas como parámetros en sistemas de reconocimiento de locutor/idioma. De este modo, se aprovecha al máximo la información proporcionada por el reconocedor de habla, al no utilizar únicamente la información de la mejor secuencia de fonemas que proporciona el reconocedor. La no gaussianidad de estas probabilidades empeora su rendimiento; por ello, Díez las transforma en cocientes y les aplica el logaritmo (PLLRs) con lo que se obtienen distribuciones más Gaussianas y siguen partiendo de la misma información.

En este contexto, se puede entender que los PLLRs sean parámetros normalizados que tienen en cuenta la diferencia entre el valor de la verosimilitud logarítmica de un modelo dado un fonema en comparación con todos los demás (Diez y col., 2013). Para calcular los parámetros PLLRs, primero se obtienen las probabilidades a posteriori acústicas por cada unidad fonética en cada trama sumando los *phone posterior* de los diferentes estados del decodificador fonético (Ecuación 2.16).

$$N_1 = \sum_{\forall s} p(m|s, t)p(m|t) \quad (2.16)$$

donde m representa la unidad fonética, t es la trama analizada y s es el estado con el que ha sido modelada la unidad fonética.

Entonces, las LLRs de cada trama t pueden ser calculadas de las probabilidades a posteriori acústicas, donde se asume una tarea de clasificación con vectores a priori planos.

$$LLR_t^m = \log \frac{p(x_t|m)}{\frac{1}{M-1} \sum_{\forall n: \neq mp(x_t|n)}} \quad (2.17)$$

Finalmente, los M LLRs por trama resultantes son colocados en un solo vector creando los parámetros PLLRs.

2.5.2. Coeficientes Cepstrales Delta Desplazados (SDCs)

Los parámetros más utilizados en los sistemas LID acústicos son los parámetros SDC (Torres-Carrasquillo y col., 2002) que, en esencia son parámetros que modelan la curva de los coeficientes cepstrales en un punto determinado en el tiempo, por lo que son capaces de manejar información de contexto de mayor duración. Ello deriva en que los SDCs modelen de mejor manera, no solamente el cambio local (derivada) sino los cambios futuros respecto del actual.

El cálculo de los SDCs se lleva a cabo especificando cuatro parámetros: (M , D , P y K) donde M especifica el número de coeficientes cepstrales que se toman del vector de características original (MFCC), D se utiliza para calcular el valor Delta y representa el número de tramas (respecto al instante actual) que serán utilizadas para calcular el Delta-Cepstral, K es el número total de ventanas Delta que se van a concatenar y P es el intervalo que se utiliza a partir del instante actual para el cálculo de los siguientes Delta-Cepstrales. La secuencia se visualiza en la Figura 2.7.

FIGURA 2.7: Cálculo del SDC para una trama de tiempo T

2.6. Fusión, Calibración y Regresión Logística Multi-clase

2.6.1. Fusión de Información en LID

Se llama fusión de datos a la combinación de diferentes fuentes de información, lo que, en el contexto de tareas de identificación de idioma se entiende como la combinación de scores obtenidos a partir de diferentes modelos para un idioma en particular. Los modelos pueden ser entrenados con diferente información acústica, diferentes técnicas de modelado o diferentes vectores de características. Lo que se espera al final del proceso de fusión es que los aciertos y errores generados por uno de los modelos puedan ser confirmados o corregidos, respectivamente, por los otros, y así entre todos.

Con la fusión se aprovecha información obtenida de diferentes fuentes, como por ejemplo, los módulos acústicos y fonotácticos ya mencionados. En función del tipo de información que se vaya a combinar, se pueden utilizar diferentes técnicas para llevar a cabo la fusión. En los modelos que generan scores que están relacionados con valores de probabilidad, normalmente se utilizan métodos de combinación lineales ó logarítmicos de asignación de pesos a cada uno de los modelos implicados en la fusión. Para otros casos, por ejemplo, cuando las salidas de los modelos se relacionan con etiquetas de clase, los métodos que utilizan un ranking o conteo, son los más utilizados.

La fusión de datos se puede llevar a cabo de dos formas: bien con los vectores de características de la información acústica o fonotáctica obtenida a la salida de la etapa Front-End en un sistema de identificación de idioma, o bien con los scores obtenidos a la salida de varios modelos de lenguaje. A la primera forma se la suele denominar fusión de entrada, donde se concatenan los diferentes vectores de características en un único vector que será procesado por el clasificador (Figura 2.8); y la segunda donde se obtiene

FIGURA 2.8: Fusión a nivel de parámetros de entrada

FIGURA 2.9: Fusión a nivel de scores a la salida de los módulos Back-End

un score final a partir de las salidas de los modelos de lenguaje, a la que suele llamarse fusión de salida (Figura 2.9).

Dado que los sistemas destinados a este fin utilizan un número muy elevado de parámetros (Wong y Sridharan, 2003), la utilización de la fusión a nivel de parámetros de entrada se hace impracticable por la elevada dimensión de los vectores. Por ello, se recurre a la opción de la fusión a nivel de los scores de cada uno de los módulos implicados en el sistema (Ramachandran, Farrell y Mammone, 2002).

Método de combinación lineal para la fusión de información

Es la técnica más comúnmente usada para la fusión de datos debido a su simplicidad y se evalúa como la suma ponderada de las salidas de cada modelo:

$$l'(x) = \sum_{1 \leq k \leq K} \alpha_k \overrightarrow{l_k(x)} + \overrightarrow{\beta} \quad (2.18)$$

donde $\overrightarrow{l_k(x)}$ es el vector logarítmico de verosimilitud del idioma k , $(\alpha_1, \alpha_2, \dots, \alpha_n, \overrightarrow{\beta})$ son los pesos calculados en el proceso de entrenamiento y $l'(x)$ es el vector logarítmico de verosimilitud fusionado.

Método de combinación logarítmica para la fusión de información

Una alternativa al método de combinación lineal es el método logarítmico. Si los pesos de alpha están restringidos entre cero y 1 y la suma es 1, entonces con el método

logarítmico también se obtiene una distribución de probabilidad a la salida. El método de combinación logarítmica se obtiene con el producto ponderado de los modelos de salida:

$$l'(x) = \prod_{1 \leq k \leq K} l_k^{\alpha_k} + \vec{\beta} \quad (2.19)$$

2.6.2. Calibración

El objetivo de la calibración es transformar los scores que proporciona el sistema (que sólo tienen significado cuando se comparan con otros scores o con un umbral) en valores con significado probabilístico por sí solos, en una escala universal que también facilita la combinación de fuentes de información.

En función de este y otros ámbitos, todo lo que envuelve a las tareas de identificación de locutor viene siendo estudiado y optimizado continuamente. Uno de los enfoques en el estudio de la identificación de locutor es la calibración de métricas de detección (Brummer y Leeuwen, 2006), (Campbell y col., 2006b) y las aprovechamos para aplicarlas en tareas de reconocimiento de idioma, considerando eso sí, que a diferencia del caso de identificación de locutor donde se manejan dos hipótesis, en el reconocimiento de idioma se deben gestionar múltiples hipótesis. Adaptar conceptos de identificación de locutor en tareas de identificación de idioma es una práctica bastante común y particularmente eficiente, como en (BenZeghiba, Gauvain y Lamel, 2009).

La medición de la calidad de la información en un sistema de reconocimiento de habla se puede realizar de dos formas: de forma directa determinando valores de entropía, o bien de forma indirecta utilizando la información para la toma de decisiones y estimando entonces las tasas de error generadas. La información puede ser expresada como la de dos componentes:

- Información relevante (IR).- Supone la existencia de información que es de interés del usuario. La medición de este contenido se denomina *discrimination* o *refinement*.
- Información no relevante (INoR).- Es la información que no es de interés del usuario, o que en caso de serlo, al no ser comprendida, ésta no puede ser usada en su beneficio.

$$info_{usuario} = IR - INoR \quad (2.20)$$

Al segundo término de esta ecuación se le llama pérdida por calibración *calibration loss* y se deduce que el calibrador estará mejor calibrado cuánto más se acerque a cero. En este contexto, el siguiente análisis busca utilizar la información entregada al usuario para medir y mejorar su calidad con el fin último de calibrar el reconocedor.

FIGURA 2.10: Sistemas LID desde el punto de vista de la calibración

Un sistema de identificación de idioma puede verse como un sistema de dos etapas consecutivas (Figura 2.10), la primera también denominada fase de *extracción* en la que se *extraen* los scores de la señal de voz y la segunda denominada fase de *presentación* cuya entrada son los scores obtenidos en la fase de *extracción* donde se calibran dichos scores mediante la elección de un umbral *hard-decision* o mediante el mapeo de los scores a tasas de verosimilitud logarítmicas *soft-decision* (Brümmer y Preez, 2006). La acción a tomar por parte del detector a la salida de la etapa de *extracción* está en función del tipo de reconocedor que se esté utilizando, pudiendo ser: dependiente de la aplicación para el primer caso e independiente de la aplicación para el segundo.

La función de costo de detección C_{DET} se define así:

$$\widehat{C}_{DET} = P_1 c_{miss} P_{miss} + (1 - P_1) C_{fa} P_{fa} \quad (2.21)$$

Donde P_1 es la probabilidad a priori del objetivo, c_{miss} el costo de falso rechazo y P_{miss} su correspondiente probabilidad.

Para los reconocedores dependientes de la aplicación que es el caso que nos ocupa, la calibración se lleva a cabo mediante la optimización de la función de costo de detección, que es una forma de cuantificar las consecuencias (falso rechazo y falsa aceptación) de las acciones tomadas por el clasificador. Entendiendo que este tipo de reconocedores dependen de un umbral que le permita al clasificador decidir entre aceptación o rechazo, la estimación de la función de costo de detección se redefine de la siguiente manera:

$$\widehat{C}_{DET}(t_{sys}) = P_1 c_{miss} P_{miss}(t_{sys}) + (1 - P_1) C_{fa} P_{fa}(t_{sys}) \quad (2.22)$$

donde t_{sys} es el umbral del score aplicado y

donde C_{miss} y C_{fa} son los costos de falso rechazo y falsa aceptación respectivamente, P_1 es la probabilidad a priori del objetivo y P_{miss} y P_{fa} son las tasas de falso rechazo y falsa aceptación respectivamente.

Nótese que:

$$0 < \widehat{C}_{DET}^{min} < \widehat{C}_{ref} \quad (2.23)$$

$$\hat{C}_{DET}^{min} < \hat{C}_{DET} \quad (2.24)$$

Donde $\hat{C}_{ref}$ es el valor de referencia del detector, que se obtiene cuando la entrada de voz es inexistente y la probabilidad a posteriori es la probabilidad a priori.

El coste de calibración (debido exclusivamente a una calibración imperfecta) es igual a:

$$\hat{C}_{DET}^{cal} = \frac{\hat{C}_{DET} - \hat{C}_{DET}^{min}}{\hat{C}_{ref}} \quad (2.25)$$

2.6.3. Regresión Logística Multi-clase

La regresión logística es un instrumento estadístico que permite predecir la probabilidad de que ocurra o no un evento determinado basándose en los valores de un conjunto de variables predictoras e independientes. La regresión logística evalúa la influencia de cada una de las variables independientes sobre el evento a predecir y controla el efecto de las demás. También define qué variables pesan más para aumentar o disminuir la probabilidad de que se produzca el evento en cuestión (Chitarroni, 2002).

En relación con tareas de LID, la regresión logística tiene como objetivo encontrar una regla de clasificación por medio de los datos de entrenamiento de tal forma que pueda asignar correctamente una etiqueta de clase a un vector de entrada de idioma desconocido (Karsmakers y col., 2007).

La regresión logística es una técnica de aprendizaje discriminativo, la que, a pesar de haber sido concebida para sistemas de dos clases, puede ser extendida a sistemas multi-clase.

El modelo de regresión logística se entrena para discriminar entre los idiomas a reconocer, sus scores pueden ser interpretados como verosimilitudes logarítmicas o como ratios de verosimilitud logarítmica, dependiendo del enfoque con el que se realice el cálculo. Para el primer caso, en el que se determina el valor de los scores s_{it} , a cada vector de entrada le corresponde un score por cada uno de los N idiomas a reconocer. Mientras que para el segundo caso se utilizan $N-1$ componentes, ya que, para determinar el ratio entre verosimilitudes logarítmicas se utilizan los parámetros de uno de los N lenguajes (generalmente el último) como elemento comparativo de todos los demás. Es evidente notar que para poder determinar la razón de cambio entre verosimilitudes logarítmicas λ_{jt} es necesario contar con los scores de verosimilitud logarítmica s_{it} (Leeuwen y Brummer, 2006). Así, para cada vector de entrada t , se calcula:

$$s_{it} = w_i * x_t + k_i \text{ Para, } i = 1, \dots, N \quad (2.26)$$

y,

$$\lambda_{jt} = s_{jt} - s_{Nt} \text{ Para, } j = 1, \dots, N \quad (2.27)$$

Donde, en la ecuación 2.26 x_t representa los scores de los vectores de entrada obtenidos a la salida de la fase de *extracción* de un sistema de LID, s_{it} son los scores de verosimilitud logarítmica resultado de aplicar los parámetros del modelo M de cada idioma a los scores de los vectores de entrada ω , λ_{jt} son los scores que representan el logaritmo del cociente de las probabilidades de los scores x_t dados dos elementos, el idioma y los parámetros del modelo cada idioma.

Estos parámetros están definidos por los elementos w_i y k_i que definen formalmente el modelo de regresión logística M , que se entrena por medio de una transformación afín aplicada al score x_t que lo lleva al espacio $LLR(N-1)dimensional$ donde N es el número de lenguajes a reconocer.

$$M = (w_1, k_1), (w_2, k_2), \dots, (w_N, k_N) \quad (2.28)$$

La diferenciación entre los scores transformados s_{jt} obtenidos para cada idioma y los scores del último idioma seguido de la extracción del logaritmo, generan los scores LLRs transformados λ_{it} , tal como se puede observar en la ecuación 2.27. Donde λ_{jt} representa la probabilidad a posteriori del score evaluado, L_j la etiqueta del idioma con el que se corresponde la expresión de entrenamiento y M el modelo de regresión logística (Leeuwen y Brummer, 2006). También se puede representar así,

$$\lambda_{jt} = \log \frac{p(x_t|L_j, M)}{p(x_t|L_N, M)}, \quad j = 1, \dots, N - 1 \quad (2.29)$$

La función básica objetivo de regresión logística se define como la probabilidad logarítmica total a posteriori de las etiquetas de los datos de entrenamiento, dados los vectores de características para estos datos. Ahora, si se asume que los scores de los vectores de entrada son estadísticamente independientes, puede escribirse de la siguiente manera:

$$\log \prod_{t \in T} P(L_t|x_t, M) \quad (2.30)$$

donde T es el conjunto de datos de entrenamiento, L_t es el idioma verdadero para el vector t y x_t es el supervector de características para ese vector de entrada. La probabilidad a posteriori $P(L_t|x_t, M)$ se puede definir en términos de los scores de regresión logística, S_{it} .

$$P(L_i|x_t, M) = \sigma_{it} = \frac{\exp s_{it} + \gamma_i}{\sum_{1 \leq j \leq N} \exp s_{jt} + \gamma_j} \quad (2.31)$$

donde $\gamma_i = \log P(L_i)$ representa la distribución de probabilidad a priori de los idiomas a reconocer y $P(L_i)$ es el cociente del subconjunto de los vectores de entrada de cada idioma T_i sobre el conjunto de los vectores de entrada de todos los idiomas T (Leeuwen y Brummer, 2006). La función objetivo que se necesita maximizar para determinar los parámetros óptimos de M es:

$$O(M) = \sum_{1 \leq i \leq N} \sum_{t \in T_i} \log \sigma_{it} \quad (2.32)$$

La aproximación adecuada de esta función objetivo posibilita el entrenamiento de un modelo de regresión logística multi-clase y define una regla de clasificación para los vectores de entrada de idioma desconocido en el sistema de reconocimiento.

Capítulo 3

MARCO DE EXPERIMENTACIÓN

3.1. Bases de datos

Se ha medido el rendimiento de los sistemas diseñados y desarrollados en la tesis sobre la base de datos KALAKA-3 desarrollada para la competición de reconocimiento de idioma Albayzin 2012 (Rodríguez-Fuentes y col., 2016) con textos en Español, Inglés, Portugués, Gallego, Catalán y Vasco, con 108 horas de voz, datos obtenidos de grabaciones de difusión de TV y con alrededor de 14000 ficheros etiquetados para el entrenamiento.

3.1.1. KALAKA-3

La base de datos KALAKA-3 utilizada en la competición ALBAYZIN 2012 LRE es el resultado de una evolución que engloba las evaluaciones realizadas anteriormente. Las bases de datos utilizadas en las competiciones ALBAYZIN 2008 LRE y ALBAYZIN 2010 LRE se han incluido como parte de KALAKA-3, añadiendo material adicional. Actualmente contiene información referente a los idiomas existentes en España (Vasco, Catalán, Español, Gallego) además de dos idiomas con fuerte influencia como el Inglés y el Portugués.

Está construida a partir de segmentos de habla obtenidos de emisiones de TV, así como de vídeos de YouTube. Las señales de habla obtenidas de la TV se han utilizado para el entrenamiento, mientras que los datos de audio extraídos a partir de vídeos de YouTube han sido utilizados para el ajuste y evaluación de los sistemas de reconocimiento.

Posee dos condiciones de experimentación: *plenty-of-training* que involucra a los seis idiomas a reconocer (Vasco, Catalán, Inglés, Gallego, Portugués y Español) y en el que se posee una gran cantidad de datos de entrenamiento y la *empty-of-training* que involucra a cuatro idiomas (Francés, Alemán, Griego e Italiano), para los que no existen datos de entrenamiento. Para permitir la evaluación de sistemas abiertos *open* (donde la clasificación se hace incluyendo a idiomas que no han participado en el entrenamiento) el conjunto

CUADRO 3.1: Estadísticas de la base de datos KALAKA-3

Ficheros	Entrenamiento	Ajuste	Eval
Núm. Ficheros	4656	458	941
Núm. Ficheros limpios	3060	-	-
Núm. Ficheros con ruido	1596	-	-
0 <duración <30 seg	2855	121	267
30 <duración <120seg	1801	337	674

de datos incluyen ficheros de habla en 11 idiomas adicionales (Búlgaro, Checo, Croata, Finlandés, Húngaro, Polaco, Rumano, Ruso, Serbio, Eslovaco y Ucraniano). KALAKA-3 contiene tres subconjuntos de datos: entrenamiento, desarrollo y evaluación.

El conjunto de datos de entrenamiento está formado por grabaciones de TV almacenadas por un único canal a 16 kHz y codificados con PCM (Pulse Code Modulation) de 16 bits en archivos WAV. Está a su vez dividido en dos subconjuntos, uno correspondiente a voz limpia (alrededor de 86 horas) y otro a voz ruidosa (alrededor de 22 horas).

Los conjuntos de datos de desarrollo y evaluación consisten en señales de audio obtenidas de vídeos de Youtube, obtenidos originalmente en diferentes formatos y calidades y almacenados a 16 kHz y codificados con PCM de 16 bits en archivos WAV. En la Tabla 3.1 se puede ver un resumen de la distribución de los ficheros. Para más información se puede consultar (Rodríguez-Fuentes y col., 2016).

3.2. Métricas utilizadas para definir el rendimiento de los sistemas de reconocimiento

3.2.1. Entropía

En general, se define la entropía como la medida de la incertidumbre que hay en un sistema, ó lo que es lo mismo, la probabilidad de que ocurra cada uno de los posibles resultados, ante una determinada situación. Así, dado un determinado conjunto de datos S , donde el atributo objetivo se corresponde con una de C diferentes clases, entonces la entropía de S relativa a dicha clasificación C se define de la siguiente manera:

$$\text{Entropía}(S) \equiv \sum_{1 \leq i \leq C} -p_i \log_2 p_i \quad (3.1)$$

donde p_i es la proporción de S perteneciente a la clase i .

Desde el punto de vista de la inteligencia artificial, si buscamos que un agente inteligente maximice su rendimiento en el tiempo, se requiere maximizar su capacidad de predecir el resultado de sus propias acciones. Si suponemos que este agente actúa en el mundo real, en entornos como los existentes en el tratamiento de lenguaje natural, se requiere que el agente tenga la capacidad de encontrar y aprovechar de los parámetros existentes en la colección de datos. En este contexto, la entropía es una métrica idónea para validar o no las capacidades de cualquier tipo de agente al tener que predecir un evento en función de los parámetros existentes en el entorno, y por tal, resulta muy útil en la evaluación de sistemas de habla donde la predicción de eventos es un factor fundamental.

3.2.2. Función de coste de detección promedio (C_{avg})

Por otra parte, el rendimiento de los sistemas de detección se ha caracterizado habitualmente por las tasas de error de falsa aceptación y de falso rechazo. En concreto, se ha utilizado una métrica que pondera los dos tipos de error y se denomina *Average Decision Cost Function* (C_{avg}) (Martin y Greenberg, 2010). Se muestra en la ecuación 3.2.

$$C_{avg} = \frac{1}{N_L} \sum_{L_T} C_{miss} P_{trgt} P_{miss}(L_T) + \sum_{L_N} C_{FA} P_{NONtrgt} P_{FA}(L_T, L_N) + C_{FA} P_{OutSet} P_{FA}(L_T, L_O) \quad (3.2)$$

donde: N_L es el número de idiomas a reconocer. L_T es el idioma objetivo y L_N representa un idioma no objetivo. L_O representa un idioma que está fuera del conjunto de idiomas a reconocer.

$$C_{miss} = C_{FA} = 1$$

$$P_{Trgt} = 0,5$$

$$P_{OutSet} = 0,0 \text{ para la condición plenty-closed.}$$

$$P_{OutSet} = 0,2 \text{ para la condición open-set.}$$

$$P_{NONtarget} = \frac{1 - P_{Target} - P_{out-of-set}}{(N_L - 1)}$$

3.3. Los reconocedores de fonemas

El componente clave para incluir información fonotáctica en los sistemas de reconocimiento de idioma es el reconocedor de fonemas, que en principio funcionan en base de HMMs para la generación de las secuencias y del algoritmo de Viterbi para el realineamiento y definición del fonema correspondiente. El objetivo del reconocedor es obtener la secuencia de fonemas y la probabilidad a posteriori de cada uno de ellos a

partir de una señal de voz. El reconocedor de fonemas de la Universidad de Brno (Schwarz, 2009) que utiliza un sistema híbrido de redes neuronales y HMMs usa modelos de tres estados por fonema, lo que significa que se tienen tres probabilidades a posteriori por unidad para cada trama. Es con este reconocedor que hemos conseguido los mejores resultados en nuestros trabajos previos y por ello lo hemos usado en el presente trabajo. Es de libre distribución e incluye modelos de 3 idiomas distintos: Húngaro, Checo y Ruso. Con el modelo de Húngaro se llega a identificar un total de 61 clases de fonemas, con el de Checo un total de 46 y con el de Ruso 53. En resumen, su funcionamiento parte de los MFCCs, a los que se les aplica una transformada discreta coseno (DCT) que sirve para de-correlacionar y disminuir la dimensionalidad propia de los MFCCs, para luego entrenar la red neuronal para producir las probabilidades a posteriori de los fonemas.

3.4. Sistema acústico de referencia

Si bien la presente tesis está orientada al desarrollo de sistemas de reconocimiento a nivel fonotáctico, también es ampliamente conocido que la fusión (Brummer y col., 2007) de éstos con sistemas acústicos (que de partida son los sistemas que mejores resultados presentan) llega a mejorar aún más las prestaciones de estos últimos. Por ello, con el fin de abordar este ámbito y de evaluar la factibilidad real del uso de la fusión de sistemas que contienen información obtenida de diferentes fuentes, llevamos a cabo la fusión del sistema acústico que se va a describir a continuación con las diferentes propuestas fonotácticas desarrolladas en esta tesis. La fusión es un procedimiento que usamos en la fase final de la etapa de clasificación, utilizando para ello, las métricas C_{avg} (Martin y Greenberg, 2010) de los sistemas fonotácticos y acústico. Se van a mostrar los resultados obtenidos en la fusión del sistema acústico con cada uno de los sistemas fonotácticos propuestos al final de la experimentación de cada una de las técnicas fonotácticas evaluadas.

Se considera que el sistema acústico de referencia utilizado tiene dos fases de procesamiento. La primera llamada Front-End, en la cual se realiza el pre-procesamiento de la señal de audio y que deriva en la creación de los supervectores. Estos se utilizan en la segunda fase denominada Back-End donde se realizan dos procesamientos consecutivos, primero la creación de los i-Vectores (Dehak y col., 2011a) y, segundo, el proceso de clasificación por medio de un clasificador de regresión logística multi-clase (MLR) (Brümmer y Preez, 2006) el cual determina el idioma con el que se corresponde la señal de audio evaluada.

En la primera fase del procesamiento, para cada fichero se extraen 12 coeficientes cepstrales Mel (MFCCs) entre los que se incluye C0. Los segmentos de silencio y de

ruido de la señal acústica se suprimen usando un Detector de Actividad de Voz (VAD). Con el fin de reducir las variaciones de ruido en las bandas de frecuencia se utiliza un filtro RASTA junto con una normalización Cepstral de media y de varianza (CMNV). Se establece para las tramas de voz una separación de 10ms y para cada una se extrae un vector de características de dimensión 56 generado a partir de la concatenación de los parámetros SDC (Torres-Carrasquillo y col., 2002) usando una configuración 7-1-3-7.

El vector generado se considera de alta dimensión y por ello suele ser llamado super-vector y en su conjunto, se utilizan para el entrenamiento, por una parte, de un modelo universal UBM, y por otra, de la matriz de variabilidad total T . Tanto el UBM como la matriz T se utilizan a su vez, para el cálculo de los correspondientes i-Vectores mediante el modelado subespacios multinomiales (Povey y col., 2010), que permite entrenar vectores de baja dimensión (i-Vectores) en el subespacio de la variabilidad total, que significa que, además de la información propia del idioma y el locutor, incluye también información de la sesión, con lo que idealmente los i-Vectores representan el conjunto de fenómenos existentes en la producción del habla.

Respecto a la dimensión que deben tener los i-Vectores para ser óptimos, tomamos en consideración los buenos resultados obtenidos por D'Haro (D'Haro y col., 2014) en una experimentación llevada a cabo sobre la misma base de datos, utilizando la dimensión de 400. Con esta dimensión también se han obtenido los mejores resultados en la mayoría de aplicaciones relacionadas con LID y por ello lo definimos con dicho valor.

Otro de los parámetros clave de los i-Vectores es el número de gaussianas a utilizar, entendiendo que mientras más gaussianas se utilicen los i-Vectores quedan mejor entrenados y por ello se debería utilizar el número máximo posible de gaussianas, pero debe tenerse en cuenta el compromiso de precisión y coste computacional. Por ello, para el tamaño de la base de datos de KALAKA-3 (Rodríguez-Fuentes y col., 2016), obtuvimos en experimentos previos que para este sistema la configuración óptima se obtiene utilizando 512 gaussianas. Entre las ventajas de los i-Vectores al trabajar con vectores de baja dimensión se encuentran la reducción de los problemas de dispersión de los datos y la mayor facilidad en el entrenamiento del clasificador. La obtención de los i-Vectores se realiza por medio del método de estimación y maximización (EM) y su optimización por medio del método de Newton-Raphson. Para mayores detalles acerca de las formulaciones matemáticas de los GMM y su aplicación en sistemas de reconocimiento de idioma se recomienda la lectura de (Kockmann y Ferrer, 2010) y (Soufifar y col., 2011).

Finalmente los i-Vectores se calibran (Brummer y Leeuwen, 2006) y utilizan como vectores de características en el entrenamiento de un clasificador multi-clase de regresión logística, que nos permite definir el idioma más probable para cada uno de los ficheros de voz evaluados. La estructura del sistema acústico se muestra en la Figura

FIGURA 3.1: Estructura del sistema acústico basado en MFCCs

CUADRO 3.2: Sistema Acústico de referencia con Coeficientes MFCCs para KALAKA-3.

	Cavg
Sistema basado en MFCCs	7,60

3.1.

La técnica se aplica sobre los datos de KALAKA-3 (Rodríguez-Fuentes y col., 2016) en su condición *plenty* y *closed*. La métrica utilizada para evaluar el rendimiento del reconocedor es y será para todos los experimentos el C_{avg} (Martin y Greenberg, 2010), que mide la cantidad de falsos rechazos y falsas aceptaciones generadas por el clasificador.

Capítulo 4

POSTERIORGRAMAS E INFORMACION DISCRIMINATIVA

4.1. Introducción

En este capítulo se presenta la unificación de dos técnicas utilizadas para el reconocimiento de idioma. Por una parte está la técnica basada en posteriorgramas (D'Haro y col., 2013) y, por otra parte, la técnica basada en información discriminativa de idioma (Caraballo y col., 2010).

La generación de los *posteriorgramas* utiliza los valores de probabilidad a posteriori de cada fonema o estado en cada trama de las señales de audio. Esta información se genera a la salida de reconocedores fonéticos, como los de la Universidad de Brno (Schwarz, 2009).

Dichas probabilidades a posteriori (*phone posterior*) se pueden utilizar como parámetros en los sistemas de reconocimiento de locutor o idioma. De este modo, se aprovecha al máximo la información proporcionada por el reconocedor de habla, al no utilizar únicamente la información de la mejor secuencia de fonemas que proporciona el reconocedor.

Por otra parte está el sistema basado en información discriminativa de idioma que se inspira en la categorización de textos propuesto con Cavnar y que ha sido mejorado mediante la aplicación de variaciones de las técnicas tf-idf, consiguiendo otorgar mayor relevancia a los n-gramas que más aparecen en un idioma y menos en los demás (D'Haro, 2009). Con la aplicación de este concepto se obtiene un ranking que es la fuente de información discriminativa con el que hemos buscado mejorar el sistema basado en posteriorgramas.

Las probabilidades a posteriori de cada uno de los estados o fonemas del reconocedor que luego se convierten en posteriorgramas mejorados gracias a la inclusión de información discriminativa se agrupan en un vector que se usa para entrenar el modelo

FIGURA 4.1: Sistema Fonotáctico basado en la obtención de i-Vectores

universal UBM correspondiente, y la matriz de variabilidad total T , elementos necesarios para la obtención de los i-Vectores. De esta manera, en cada instante de tiempo o trama de voz se tiene un vector con los posteriorgramas mejorados que, junto con los otros obtenidos en una frase, se utilizan para obtener un i-Vector.

Los i-Vectores obtenidos se utilizan finalmente como entradas de un clasificador discriminativo multi-clase de regresión logística con el fin de obtener la función de coste de detección promedio C_{avg} (Martin y Greenberg, 2010) que mide la bondad del clasificador en función de la ponderación de falsos rechazos y de falsas aceptaciones generadas en los ficheros evaluados. El sistema referenciado se puede ver en la Figura 4.1.

4.2. Sistema LID basado en posteriorgramas

La primera técnica basada en información fonotáctica utilizada en este trabajo ha sido la creación de posteriorgramas (D'Haro y col., 2013).

Los posteriorgramas son vectores de características creados a partir de las probabilidades a posteriori de cada uno de los fonemas de las secuencias obtenidas como salida del reconocedor de fonemas (Schwarz, 2009). Siendo más específicos, los posteriorgramas son probabilidades condicionales de ocurrencia de un determinado fonema considerando la información aportada por los $n-1$ fonemas previos. Los posteriorgramas contienen información a nivel de trama, para cada una de las cuales se genera un vector. Son estos vectores los utilizados para el entrenamiento del UBM y de la matriz de variabilidad total de donde se extraen los i-Vectores (D'Haro y col., 2013).

La creación de posteriorgramas se describe en los siguientes pasos:

FIGURA 4.2: Procedimiento de creación de los posteriorigramas

Para comenzar con la creación de los posteriorigramas, es necesario extraer los valores de probabilidad a posteriori de cada uno de los posibles fonemas a reconocer para cada trama. Estos valores se obtienen directamente del reconocedor de fonemas.

En la Figura 4.2 se puede ver la creación de posteriorigramas mediante un ejemplo.

En un primer paso, para cada trama del fichero de audio se extraen 3 valores correspondientes a los 3 posibles fonemas a reconocer (hemos usado 3 fonemas para simplificar el ejemplo, ya que en el sistema real fueron 33 fonemas distintos).

En un segundo paso se suman y promedian las probabilidades a posteriori de todas las tramas que se consideran que pertenecen a la misma unidad fonética. Esta agrupación de las tramas en fonemas se realiza también con el reconocedor de fonemas empleando el algoritmo de Viterbi sobre las probabilidades a posteriori obtenidas en el paso primero.

El tercer paso consiste en calcular las probabilidades condicionales de que ocurra un determinado fonema considerando los $n-1$ fonemas previos (los n -gramas). Para ello, en el caso de usar bigramas, como se muestra en la Figura 4.2, se realiza el producto exterior (outer-product) entre las probabilidades promediadas del fonema previo con las del fonema actual. Para solventar el problema de la primera trama se ideó un fonema

CUADRO 4.1: Sistema Fonotáctico de referencia basado en i-Vectores usando posteriorgramas.

	Cavg %
Sistema basado en posteriorgramas	9,85

tipo *dummy* en el que todos los fonemas son equiprobables.

El cuarto paso consiste en sumar todas las matrices producto generadas antes a lo largo de todo el fichero cuidando de sumar adecuadamente los mismos contextos (la probabilidad condicional $p_{ij}(t-1)$ con la probabilidad $p_{ij}(t)$). El resultado es lo que denominamos posteriorgramas condicionales.

Por último se convierte la matriz de posteriorgramas en un vector de dimensión $[F^n \times 1]$, donde F es el número de fonemas a emplear y n es el orden de los n-gramas. En nuestro caso, al emplear 33 fonemas y usar bigramas se obtiene un vector de dimensión 1089, y en el caso de trigramas un vector de dimensión 35937. Estos vectores se crean para cada fichero y para cada uno de los idiomas a reconocer, siendo luego utilizados para la obtención de los i-Vectores.

El sistema ha sido configurado para ser evaluado sobre los datos de KALAKA-3 en la condición *plenty, closed* extrayendo sus correspondientes i-Vectores de dimensión 400. La función de coste de detección Cavg (Martin y Greenberg, 2010) obtenida se puede ver en la Tabla 4.1.

4.3. Sistema de rankings de información discriminativa

Hemos considerado que la frecuencia de ocurrencia de las unidades fonéticas y su correspondiente categorización se puede utilizar como parámetro en un sistema de reconocimiento de idioma, por lo que se han explorado técnicas para introducir esta información en el sistema. Así encontramos que la categorización de texto propuesta por Cavnar (Cavnar y Trenkle, 1994) y la generación de rankings discriminativos propuesto por D'Haro (D'Haro, 2009) se adaptaban perfectamente a las condiciones de experimentación de la técnica de posteriorgramas, y por ello, decidimos combinarlas con el fin de mejorar el rendimiento del reconocedor.

Si bien la frecuencia de ocurrencia es un parámetro de mucha utilidad para definir la importancia de un n-grama en un idioma, también presenta desventajas, como el hecho de que no brinda información relativa a la frecuencia de aparición de ese mismo n-grama en otros idiomas. Por ello, en (D'Haro, 2009) se propuso la técnica de utilizar rankings

discriminativos, basándose en dar más relevancia a los n-gramas que aparecen más en un idioma y menos en los demás.

En ese contexto, y siguiendo el método descrito en el apartado 2.4.5, creamos el ranking discriminativo a nivel de n-gramas.

El sistema de generación de rankings de información discriminativa conlleva un procesamiento supervisado donde se entrenan modelos de lenguaje que son representados por medio de rankings que contienen, por un lado, al n-grama del texto entrenado y, por otro, su valor discriminativo.

En la fase de evaluación para cada fichero se sigue lógicamente el mismo procedimiento. Utilizando la secuencia de n-gramas generada por el reconocedor fonético, se crea un ranking con los n-gramas y su valor discriminativo que se compara con cada uno de los rankings entrenados para cada idioma.

La decisión del reconocedor para elegir el idioma con el que se corresponde el texto de evaluación se basa en calcular la diferencia en las posiciones en el ranking entre los n-gramas que aparecen en la frase a evaluar y los que aparecen en los rankings entrenados. Si el ranking generado a partir del texto de evaluación contiene muchos de los n-gramas existentes para un ranking entrenado, la métrica de distancia será menor y tenderá a ser el idioma elegido. Esta técnica de *ranking* puede considerarse una forma de comparar los textos de entrenamiento y de evaluación y se corresponde con la diferencia absoluta entre las posiciones de los n-gramas de los dos rankings (Cordoba y col., 2007). Por ejemplo, aquel ranking generado a partir de una señal de audio de evaluación con la que se presente una menor distancia al compararla con el *ranking* del conjunto de datos de entrenamiento de un idioma en particular, indicará una mayor coincidencia de n-gramas y el idioma al que pertenezca dicho ranking de entrenamiento corresponderá al idioma seleccionado por el reconocedor. La distancia se calcula con la ecuación 4.1:

$$d^T = \frac{1}{N} \sum_{1 \leq t \leq N} \text{abs}(\text{pos}w_i - \text{pos}w_i^T) \quad (4.1)$$

Donde N es el número de n-gramas generados para la frase a reconocer, $\text{pos}w_i$ es la posición del término w_i en el ranking de entrenamiento, en tanto que $\text{pos}w_i^T$ es la posición del mismo término w_i localizado en el fichero de evaluación.

Como ejemplo, el ranking que se presenta en la Tabla 4.2 se obtiene sobre los datos de KALAKA-3, para el HMM de Húngaro y el idioma Catalán, y contiene la siguiente información:

- La definición del tipo de n-grama (bigrama ó trigrama).- En la Tabla 4.2 se muestran dos ejemplos, uno para bigramas donde el ranking limita a los valores discriminativos entre 0 y 1, y su codificación numérica (31 y 10) y otro para trigramas donde los valores discriminativos del ranking están limitados entre -1 y 1, junto con su codificación

CUADRO 4.2: Formato del ranking utilizado.

	n-gram	V. discriminativo	Frec.Ocurrencia
bigrama	31-10	0,106227	3771
trigrama	06-27-01	-0,307244	20

numérica (06, 27 y 01).

- El valor discriminativo (V. discrim). Este valor define la relevancia del n-grama para el idioma propio frente al resto de los idiomas.

- La frecuencia de ocurrencia (Frec. ocurrencia).- Este factor describe la cantidad de veces que el fonema ha aparecido en el segmento de voz analizado.

Los rankings en sí mismos son elementos capaces de otorgar discriminabilidad a un clasificador (Caraballo y col., 2010), ya que, para cualquier momento se cuenta con información de la importancia que puede tener un n-grama para uno y otro idioma y ésta puede ser utilizada para potenciar sistemas más complejos de reconocimiento que utilicen unidades fonéticas similares. El valor de discriminabilidad $n1''$ de cada n-grama se calcula con una variante de la técnica tf-idf (D'Haro, 2009) por medio de la ecuación

4.2:

$$n1''(\omega) = n1'(\omega) \frac{(n1'(\omega) - n2'(\omega))}{n1'(\omega) + n2'(\omega))^2} \quad (4.2)$$

Donde la diferencia $n1'(\omega) - n2'(\omega)$ está constituida por $n1'(\omega)$ que es el número de ocurrencias en el idioma actual y por $n2'(\omega)$ que es el número promedio en los otros idiomas. El denominador $n1'(\omega) + n2'(\omega))^2$ junto con el factor multiplicativo $n1'(\omega)$ son valores de normalización (Caraballo y col., 2010) que provocan que el valor discriminativo esté comprendido entre -1 y 1.

De los datos que se obtienen en los rankings hemos utilizado únicamente las columnas relacionadas con el nombre del n-grama y su valor discriminativo.

4.4. Incorporación de información discriminativa en los posteriorgramas

Sobre la base de los sistemas descritos en los apartados 4.2 y 4.3, nos propusimos mejorar el rendimiento del sistema fonotáctico de reconocimiento basado en posteriorgramas aprovechando el sistema de generación de rankings discriminativos de n-gramas.

Para ello, hemos incluido los valores discriminativos en los posteriorgramas, que se obtienen a la salida del reconocedor de fonemas, luego del proceso de creación de los posteriorgramas, ya que en ese punto contamos con la información del n-grama analizado y de su valor de cuenta de posteriorgrama condicional.

En la práctica, lo que se hace es localizar el n-grama generado por el reconocedor de fonemas en los rankings discriminativos y se añade su valor al correspondiente valor de cuenta de posteriorgrama condicional. El objetivo de incluir dicho valor es que si un n-grama es muy discriminativo para un idioma y menos para los otros, se busca que la probabilidad de ocurrencia del mismo aumente, ya que la probabilidad de que dicho n-grama pertenezca a ese idioma aumenta. En caso contrario, si el n-grama es poco discriminativo para su idioma, entonces se reduce el valor de la cuenta de posteriorgrama condicional, ya que habrá menos probabilidad de que dicho n-grama pertenezca a ese idioma.

Para ello, lo que hacemos es multiplicar el valor del posteriorgrama por el valor discriminativo obtenido en el ranking, utilizando la ecuación 4.3. La evaluación se realiza a nivel de trigramas.

$$C_n^i = (1 + \omega_n^i) * Co_n^i \quad (4.3)$$

Donde C_n^i es la cuenta de posteriorgrama objetivo en el idioma i , Co_n^i es la cuenta de posteriorgrama original en el idioma i y ω_n^i es el valor discriminativo extraído del ranking para el idioma i , con un valor comprendido entre -1 y 1. Por su parte, n representa el n-grama analizado dentro de una secuencia de n-gramas obtenida como salida del reconocedor. El factor de multiplicación queda comprendido entre 0 y 2, por lo que lógicamente nunca se hace cero.

El procedimiento descrito se puede ver en la Figura 4.3.

Los valores de posteriorgramas condicionales modificados, los usamos luego de manera similar a lo propuesto en el sistema basado en posteriorgramas, es decir, se concatan en cada fichero para formar los respectivos vectores de características, con los que se entrenan, el UBM y la matriz de variabilidad total T , a partir de los cuales se obtienen los i-Vectores (Dehak y col., 2011a) que luego se calibran y fusionan para definir por último la función de coste de detección promedio C_{avg} (Martin y Greenberg, 2010) que es la métrica con la que hemos evaluado esta técnica.

El resultado de la aplicación de la técnica se puede ver en la Tabla 4.3.

La mejora relativa del 8,63 % en C_{avg} nos animó a explorar otros ámbitos en el proceso de inclusión de información discriminativa, entre los que destacamos:

- La reducción del tamaño de los rankings en función del número de apariciones de un n-grama en los ficheros evaluados, y,

FIGURA 4.3: Incorporación del ranking discriminativo a los posteriorgramas

CUADRO 4.3: Efectos de incluir información discriminativa en el sistema basado en posteriorgramas.

	C_{avg}	Mejora %
Posteriorgrama (Ref)	9,85	
Posteriorgrama (Ref) + Ranking	9,00	8,6

- La inclusión de un factor de suavizamiento en la fórmula utilizada para la transformación de los valores de posteriorgramas.

Respecto al tamaño de los rankings, en especial las de n-gramas de mayor orden, de 3 en adelante, es común tener un gran porcentaje de n-gramas que han aparecido muy pocas veces, con lo que se incrementa considerablemente el coste computacional sin que mejore la tasa de reconocimiento. Por ello, evaluamos el sistema limitando el tamaño de los rankings aplicando un umbral al número mínimo de ocurrencias de los n-gramas a considerar.

Finalmente, proponemos incluir un factor de suavizamiento en la Fórmula 4.3. Observamos que el resultado de la fórmula generaba en ocasiones cambios excesivos en la cuenta del posteriorgrama decidimos aplicar un factor de escala en el valor discriminativo, de tal manera que se suavizara su impacto sobre la cuenta de posteriorgrama. Para ello hemos aplicado un factor al que hemos llamado *de suavizamiento*.

Estas mejoras se han abordado en el orden siguiente: en primer lugar se ha aplicado el factor de suavizamiento, tanto en bigramas como en trigramas y luego, evaluar el efecto de la reducción del tamaño de los rankings mediante un valor de umbral.

4.4.1. Factor de suavizamiento en los posteriorgramas discriminativos

Como acabamos de comentar, decidimos introducir un factor de suavizamiento α (experimentando con valores entre 0 y 1) multiplicándolo por el valor discriminativo ω_n^i en la ecuación 4.3. De esta manera, en las condiciones límites, con un valor de $\alpha = 0$, el impacto del valor discriminativo es nulo, en tanto que con un valor de $\alpha = 1$, el valor discriminativo íntegro se suma a la constante unitaria y el resultado se multiplica por la cuenta de posteriorgrama. Este factor de suavizamiento se ve reflejado en la ecuación 4.4:

$$C_n^i = (1 + \alpha \omega_n^i) * Co_n^i \quad (4.4)$$

Dados los largos tiempos de procesamiento cuando se utilizan trigramas como unidades fonéticas en esta técnica, se decidió realizar las pruebas a nivel de bigrama para escoger el mejor sistema que luego sería el utilizado en un sistema de trigramas.

4.4.2. Resultados para bigramas

Al evaluar bigramas, se obtuvo una mejora generalizada para la mayoría de factores de suavizado, obteniendo el mejor rendimiento del sistema para un factor de suavizamiento comprendido entre 0.3 y 0.4, tal y como se muestra en la Tabla 4.4.

CUADRO 4.4: Factor de suavizamiento para bigramas.

α	Cavg	Mejora %
0	11,04	
0,2	10,5	4,9
0,3	10,4	5,8
0,4	10,5	4,9
0,5	10,6	3,4
0,7	10,8	2,2

CUADRO 4.5: Factor de suavizamiento para trigramas.

	α	Cavg	Mejora %
Posteriorgrama (Ref)	0	9,85	
Ref+Ranking	0	9,00	8,6
Ref+Ranking	0,2	9,26	5,9
Ref+Ranking	0,3	9,12	7,4
Ref+Ranking	0,4	8,97	8,9
Ref+Ranking	0,5	8,95	9,1
Ref+Ranking	0,6	9,04	8,2

4.4.3. Resultados para trigramas

Tras la evaluación con bigramas, procedimos a la evaluación con trigramas. Los resultados obtenidos se muestran en la Tabla 4.5, donde se puede ver que los resultados no llegaron a ser del todo satisfactorios. Para casi todos los valores de α los valores de C_{avg} tendieron a ir peor, excepto para valores entre 0,4 y 0,5 donde la técnica mejoró (aunque no sustancialmente) los resultados de la técnica original.

4.4.4. Rankings discriminativos y el umbral mínimo de ocurrencia de los n-gramas

Al analizar las razones por las cuales no se obtuvieron mejoras sustanciales con trigramas, pensamos que se podía deber al umbral del número de ocurrencias que debe tener un n-grama de cara a incluirlo o no en el ranking por ser poco relevantes. Hasta ahora, se han descartado los n-gramas con menos de tres apariciones en la base de datos de entrenamiento, lo que implica descartar entre un 30 o 40 % del tamaño total del ranking.

CUADRO 4.6: Cambio en el umbral mínimo de ocurrencia para trigramas.

	α	Cavg	Mejora %
Posteriorgrama (Ref)		9,85	
Ranking umbral 3		9,00	8,6
Ranking umbral 0	0,4	8,96	9,0
Ranking umbral 0	0,45	8,76	11,1
Ranking umbral 0	0,5	8,82	10,4

CUADRO 4.7: Resultados de fusionar el módulo acústico con el sistema fonotáctico de referencia y el basado en información discriminativa.

	C_{avg}	Mejora %
Acústico	7,60	
Posteriorgrama (Ref)	9,85	
Ranking	8,76	
Ref + Acústico	5,69	
Ranking + Acústico	5,21	8,43

Se decidió reducir este umbral para descartar un número de n-gramas más reducido e incluso incluirlos todos utilizando un umbral 0. Finalmente, se obtuvo el resultado óptimo para dicho umbral 0, resultados que pueden observarse en la Tabla 4.6.

Como se puede ver en la Tabla 4.6 el hecho de incluir todos los n-gramas sin descartar ninguno aunque su frecuencia de aparición fuese reducida resultó beneficiosa para el sistema. Se han obtenido mejoras de C_{avg} de hasta del 11 % comparándolo con el sistema de Referencia que utiliza posteriorgramas y de hasta el 2,3 % en relación al sistema de partida con umbral 3.

4.4.5. Fusión de los posteriorgramas discriminativos y el sistema acústico de referencia

Finalmente, tomamos el mejor sistema obtenido para realizar su fusión con el sistema acústico de referencia que hemos descrito en la sección 3.3.

El resultado se puede ver en la Tabla 4.7 donde comprobamos que los sistemas fonotácticos mejoran las prestaciones del sistema acústico (7,60 %) con una mejora del 31 % gracias a la fusión. En este caso concreto, vemos cómo los resultados de la fusión del sistema que incorpora información discriminativa con el módulo acústico son mejores que los de la fusión del sistema de referencia basado en posteriorgramas con el acústico.

4.4.6. Conclusiones y aportaciones

La línea de investigación propuesta para incorporar información discriminativa de idioma en el sistema de reconocimiento basado en posteriorgramas resultó satisfactoria, ya que en su primera fase obtuvo una mejora de 8,43 % en C_{avg} . En una segunda fase, en la búsqueda de mejorar esta tasa, hemos propuesto la inclusión de un factor de suavizado y la inclusión de ngramas de poca ocurrencia. Con estas dos propuestas también encontramos resultados satisfactorios, llegando a obtener una mejora relativa de hasta el 11,1 % en C_{avg} . El sistema basado en posteriorgramas y el sistema basado en valores discriminativos de idioma son los sistemas de base sobre los cuales se realizó la aportación científica que consistió en la unificación de los dos sistemas y que ésta genere una mejora en la tarea de reconocimiento.

Capítulo 5

PARÁMETROS SDPCs PARA TAREAS LID

5.1. Introducción

En este capítulo se presenta una técnica donde se utilizan los parámetros PLLR para convertirlos en SDPCs por medio de la aplicación de la técnica SDC sobre los valores de los parámetros PLLR en vez de sobre los parámetros acústicos. Hasta el momento los PLLR se han usado a nivel de fonemas y lo que proponemos es utilizarlos a nivel de estados con el fin de aprovechar la información existente en las transiciones entre estados.

El sistema de base para la experimentación, tiene una estructura similar a la utilizada para el sistema acústico de referencia presentado en la sección 3.4.

Los parámetros *PLLR* utilizan los valores de probabilidad a posteriori de cada fonema o estado en cada trama de las señales de audio. Esta información se genera a la salida de los reconocedores fonéticos.

Dichas probabilidades a posteriori (*phone posterior*) las usamos como parámetros en los sistemas de reconocimiento de locutor o idioma.

Las probabilidades a posteriori de cada uno de los estados o fonemas del reconocedor en cada instante se agrupan en un vector, que se usa luego para entrenar un modelo universal UBM y la matriz de variabilidad total T , para obtener los i-Vectores. En cada instante de tiempo o trama de voz se tiene un vector con las probabilidades a posteriori.

Los i-Vectores entrenados se utilizan finalmente como entradas de un clasificador discriminativo multi-clase de regresión logística con el fin de obtener el Cavg que mide la bondad del clasificador en función de la ponderación de falsos rechazos y de falsas aceptaciones generadas en los ficheros evaluados. El sistema referenciado se puede ver en la Figura 5.1.

FIGURA 5.1: Sistema SDPC basado en i-Vectores

5.2. Sistema de LID basado en Shifted Delta PLLR Coefficients (SDPCs)

5.2.1. Descripción

Un elemento esencial en los sistemas de reconocimiento de idioma son los parámetros utilizados para representar la información acústica de entrada. Normalmente se obtienen a partir del espectro de frecuencias y pueden ser de varios tipos. Los utilizados en esta tesis han sido: MFCCs (Davis y Mermelstein, 1980) y PLLRs (Díez y col., 2012). A partir de estos últimos, en (D'Haro y col., 2014) proponemos los SDPCs, que son coeficientes que se crean a partir de los PLLRs. La aplicación de la técnica SDC en los PLLRs consigue mejorar el tratamiento de su contexto temporal, ya que se amplía el contexto de información y se utiliza información generada en instantes anteriores y posteriores.

Así mismo, en (D'Haro y col., 2014) recuperamos información de la verosimilitud obtenida en cada estado del reconocedor, a diferencia de la propuesta original de PLLRs, presentada por (Diez y col., 2013), en la que se promediaban las verosimilitudes de cada estado de un fonema en un único componente para ser utilizado a nivel de fonema. Con ello aprovechamos de mejor manera la información existente en las transiciones entre estados. Para agregarle información temporal a los parámetros PLLR de estado se propuso expandir la información de contexto aprovechando la filosofía de los SDCs en la que con un esquema similar al de la Figura 2.7 se le agrega información pasada y futura a los PLLRs de tal manera que se consigue generar información fonética de contexto más amplia a nivel de trama favoreciendo el reconocimiento de idioma.

Aparecen entonces los SDPCs que se utilizan como parámetros para el reconocimiento. Se obtienen a partir de los PLLRs aplicando las fórmulas de SDC a los valores de los

parámetros PLLR y su objetivo es contrarrestar las deficiencias presentadas por estos últimos a la hora de modelar un contexto de mayor duración.

5.2.2. Sistema de partida basado en los parámetros SDPCs

En este sistema, que sigue la estructura definida en el apartado 4.1, se han utilizado como vectores de características para el entrenamiento de los i-Vectores a los coeficientes SDPCs. Estos coeficientes se crearon a partir de los coeficientes PLLRs descritos por su parte en el apartado 2.5.1, y que se obtienen a partir de la probabilidad a posteriori de los fonemas generada por el reconocedor fonético respectivo.

Los coeficientes PLLRs obtenidos a nivel fonético fueron transformados en coeficientes SDPCs por medio de la aplicación de la filosofía SDC consiguiendo que los PLLRs fueran capaces de modelar información de mayor contexto y mejorar las prestaciones del sistema. Los resultados de estas mejoras se pueden encontrar en (D'Haro y col., 2014).

Además de los SDPCs basados en fonemas, también se generaron SDPCs basados en los estados de cada fonema. Así, se recuperaron los PLLR de estado generados por el reconocedor y se utilizaron para obtener los SDPCs mediante la aplicación del algoritmo SDC (Torres-Carrasquillo y col., 2002).

La desventaja que conlleva la generación de los SDPCs, tanto si se generan a partir de información de estado como si se generan a partir de información de fonema, es el tamaño de los vectores a procesar, en especial para los SDPCs de estado, en el que se triplica la dimensión del vector al tener 3 estados por fonema. Por ello, se utilizan técnicas de reducción de dimensionalidad (Duda, Hart y Stork, 2012) como PCA, LDA o HLDA con lo que se disminuye el número de parámetros en el vector manteniendo las características de representatividad de idioma. En pruebas preliminares, se comprobó que PCA (Richardson, 2009) es la técnica con la que se obtienen los mejores resultados (D'Haro y col., 2014).

En el sistema de referencia para la experimentación se realizan los pasos que se describen a continuación.

Con los modelos de los reconocedores de fonemas ya entrenados, a la salida de éstos se obtienen las probabilidades a posteriori de cada estado fonético del vocabulario que se utilizan para generar los correspondientes coeficientes PLLR. Cada uno de los coeficientes PLLR pasa a ser un elemento del vector x_k donde k es cada una de las tramas de la señal de audio procesada. Para el caso de los PLLRs de fonemas, el procedimiento es similar, únicamente cada parámetro es el promedio de los PLLRs de sus 3 estados.

En una segunda fase, con el fin de reducir la dimensionalidad de los vectores de los parámetros PLLRs (ya sea de estados o de fonemas) se aplicó PCA (Principal Component Analysis) al vector de parámetros PLLR para facilitar el entrenamiento posterior del

FIGURA 5.2: Obtención de los coeficientes SDPCs

sistema, buscando conservar las características más representativas de los PLLRs a nivel de estados o de fonemas pero en una dimensión más reducida.

Ya en una tercera fase, se toma el vector obtenido a la salida de PCA y se procede a añadirle información temporal aplicando el algoritmo SDC. De esta manera se obtienen los coeficientes SDPCs que se utilizan luego para el entrenamiento de los i-Vectores. La dimensión final del vector tras aplicar SDC es igual a la dimensión de PCA que se haya elegido multiplicada por el orden $K+1$ de SDC. En la Figura 5.2 se muestra el procedimiento descrito.

El vector con los coeficientes SDPCs generados se ha utilizado de la forma clásica de los sistemas basados en i-vectores para entrenar un modelo UBM que contiene características propias de cada uno de los idiomas a reconocer y para entrenar la matriz de variabilidad total T . Hemos usado la ecuación 2.12, seguida de manera iterativa, tras lo que se generan los i-Vectores de las secuencias fonéticas de cada uno de los ficheros de audio. Los i-Vectores se utilizan posteriormente en un clasificador que funciona mediante regresión logística y que se encarga de decidir el idioma con el que se corresponde una secuencia fonética dada. En la Figura 5.1 se representa el sistema base de experimentación descrito.

El hecho de que los SDPCs incorporen información temporal nos da la posibilidad de controlar la cantidad de contexto a incluir en cada instante de tiempo. El método utilizado fue la modificación de los parámetros de configuración de los SDPCs, que son los siguientes:

- M especifica el número de coeficientes que se van a calcular y se refiere al número de parámetros MFCC, PLP o PLLR en nuestro caso, que se van a utilizar. Cada uno de estos coeficientes se trata de forma separada. Tras aplicar PCA, M será la dimensión reducida de PCA.

- D es el valor de la diferencial o derivada que se va a utilizar para el cálculo de los coeficientes Delta y representa el número de coeficientes cepstrales que serán utilizados para formar el coeficiente Delta-Cepstra. Es el tamaño de la ventana temporal utilizada para calcular los parámetros Delta.

- P es el número de tramas que se van a aplicar entre el cálculo de un Delta-Cepstra y el siguiente y representa la longitud del salto temporal que se va a dar.

- K es el número total de coeficientes Delta que se van a concatenar para obtener los coeficientes Delta-Cepstra y representar el número total de saltos que se dan.

Requerimos optimizar varios parámetros a la vez, parámetros que tienen un efecto determinante en la dimensión final del vector y , donde debemos procurar que que no haya un sobre-entrenamiento en la obtención de los i-Vectores, dado que el tamaño de la base de datos es fijo. Si queremos un contexto temporal grande el vector tendrá una dimensión elevada, probablemente excesiva, lo que provocará un sobre-entrenamiento.

Se han realizado múltiples experimentos para encontrar una combinación óptima de valores para los parámetros M , D , P y K , incluso en (Kohler y Kennedy, 2002) se establecen las bases para evaluar la viabilidad de una técnica que calcule de forma automática los valores óptimos para estos parámetros.

En la búsqueda de estos valores óptimos fue especialmente relevante el estudio del parámetro K de los coeficientes SDC. Nos interesaba que los sistemas PLLR óptimos tuviesen un valor para el parámetro K alto, ya que esto implicaría un contexto más amplio en el cálculo del vector en un instante dado, pero siendo conscientes de que un valor de K elevado implicaba un incremento de la dimensión final del vector, lo cual terminaría provocando un peor entrenamiento para un tamaño fijo de la base de datos por el problema del sobre-entrenamiento. La dimensión final del vector de parámetros es igual al producto de $M(K + 1)$, así que fue importante buscar un equilibrio entre ambos factores. Con la elección de un M alto, la proyección PCA funcionaría mejor pero el K elegido óptimo tendería a ser más pequeño, limitando el contexto de cálculo del vector, lo que nos obligó a una solución de compromiso (Muñoz Igual, 2015).

Además de la optimización de los parámetros D , P y K de los SDCs, fue necesario evaluar diferentes dimensiones de PCA en el proceso de reducción de dimensionalidad, ya que con ello, además de controlar los efectos de la inclusión de información contextual en la tarea de reconocimiento, podemos controlar el coste computacional y el tiempo de ejecución. La configuración PCA y SDC de base que se muestra en la Tabla 5.1 fue definida considerando los resultados de D'Haro en (D'Haro y col., 2014).

El análisis se centró en utilizar el modelo de Húngaro del reconocedor, tanto para fonemas como para estados. Los resultados por su parte se evaluaron con el C_{avg} obtenido. Ya al final de todo, después de obtener los parámetros óptimos de M , D , P y K hemos incluido los resultados obtenidos al utilizar los modelos de Ruso y de Checo.

En resumen, la optimización de los SDPCs se realizó analizando el efecto producido en el sistema por:

- La reducción de dimensionalidad por medio de PCA, y

CUADRO 5.1: Configuración de partida para el sistema SDPC de fonemas y estados.

Sistema de partida	Reco	PCA	D	P	K	C_{avg} %
Fonemas	Húngaro	30	1	5	5	6,74
Fonemas	Checo	30	1	5	5	7,86
Fonemas	Ruso	30	1	5	5	7,11
Estados	Húngaro	35	1	5	3	6,57
Estados	Checo	35	1	5	3	7,33
Estados	Ruso	35	1	5	3	7,00

- La modificación de las características temporales del SDC en los SDPCs.

5.2.3. Modificación del parámetro D

El primer punto de análisis consistió en determinar el número de coeficientes cepstrales utilizados para formar el coeficiente Delta-Cepstral, la ventana temporal usada para determinar el valor Delta.

-*Para fonemas.* Al parámetro D se le asignó inicialmente el valor de base (Tabla 5.1) y luego se evaluó el sistema incrementándolo una unidad. Los otros parámetros, es decir, P y K , no se cambiaron. Para el barrido de la configuración PCA, se ha evaluado el sistema para valores de 20, 25, 30 y 35, encontrando un mínimo de C_{avg} con 25 de dimensión y $D=2$.

-*Para estados.* Para este caso, la configuración fue similar a la utilizada para fonemas, con la única diferencia de que la dimensión de PCA fue evaluada para valores entre 25 y 40 en intervalos de 5. Este nuevo dominio se eligió porque el valor de base de PCA mostrado en la Tabla 5.1 para estados era 35 y el óptimo se buscó en torno a ese valor base. Así, la configuración óptima de PCA terminó coincidiendo con la del sistema de partida y el parámetro D igual a 2.

5.2.4. Modificación del parámetro K

En relación al parámetro K , que determina el número total de coeficientes Delta que se concatenan de acuerdo al número total de saltos que se dan, el factor principal a tener en cuenta es la dimensión final del vector dado que para un número de saltos elevado es muy probable que se produzca un sobre-entrenamiento. Además, el tiempo de cálculo también se dispara.

-*Para fonemas.* La reducción de dimensionalidad se evaluó para 5 casos diferentes entre 20 y 35 dado que están cercanos a los valores de base, mientras que K se evaluó para valores entre 3 y 7 (como tope por el incremento excesivo de la dimensión del vector). Se obtuvo el mínimo con $K=4$ con una dimensionalidad de 25.

-*Para estados.* En este caso, la reducción de dimensionalidad se evaluó para 7 casos diferentes entre 25 y 70, dado que partimos de un vector de partida con una dimensión triple del caso anterior y no deseamos perder demasiada información al aplicar PCA. El valor de K varió entre 2 y 5, con los mismos criterios considerados para el caso de fonemas. El mínimo se obtuvo para $K=3$ y $K=4$ al utilizar una dimensión de 50 para el PCA, en tanto que los resultados empeoraron considerablemente para valores de dimensionalidad más altos. Por estos motivos, se han considerado como óptimos $M=50$ y $K=3$.

5.2.5. Modificación del parámetro P

El tercer y último parámetro que modificamos es el parámetro P , que es el número de tramas que se saltan entre el cálculo de un Delta-Cepstra y el siguiente, es decir, la longitud del salto.

Los valores propuestos para P han estado en torno al valor base igual a 5 (Tabla 5.1), ya que valores mucho más pequeños suponen que el contexto analizado se acorta en exceso y valores mucho mayores hacen que se abarque un contexto más amplio, pero excesivo y menos representativo de los parámetros PLLR en el instante actual, con lo que el reconocimiento empeora. Por ello, se ha evaluado el valor de P para $P=4$ y $P=6$.

-*Para fonemas.* Así, para el sistema SDPC basado en fonemas, la combinación $K=4$, $P=5$ resulta mejor que la combinación $K=5$, $P=5$ y las dos son mejores que los resultados obtenidos con $P=4$ y $P=6$. Por estas razones se consideró a la combinación $K=4$, $P=5$ como la óptima con dimensionalidad de 25.

-*Para estados.* En este caso, se obtuvo un óptimo para $P=3$ ó $P=4$ cuando la dimensionalidad es 70.

De esta manera, después de haber optimizado los valores de M , D , P y K hemos podido establecer una configuración estándar como la que se muestra en la Tabla 5.2. En ella se puede apreciar que el parámetro D es el más estable con valor igual a 2, en tanto que P puede configurarse con un valor de 5, en ambos casos tanto para fonemas como para estados. Respecto al parámetro K , el óptimo es igual a 5 para fonemas en tanto que para estados el valor es más variado y dependiente del valor de M utilizado, obteniendo óptimos para valores entre 2 y 5.

En base a estos óptimos se evaluó el sistema LID utilizando los otros modelos del reconocedor de fonemas (Ruso y Checo), llegando a obtener las siguientes mejoras en

CUADRO 5.2: Configuraciones óptimas del SDC en la creación de los SDPCs.

HMM	Fonemas			Estados		
	D	P	K	D	P	K
Ruso	2	4/6	4/5	2	5	5
Húngaro	2	5	4/5	2	3/5	3
Checo	2	5	5	2	5	2

CUADRO 5.3: C_{avg} para cada una de las configuraciones - Caso fonemas

	Sistema	C_{avg}	Mejora %
Fonemas	Húngaro PCA30+SDC 1 5 5	6,98	
Fonemas	Húngaro PCA25+SDC 2 5 5	6,62	5,16
Fonemas	Checo PCA30+SDC 1 5 5	7,86	
Fonemas	Checo PCA25+SDC 2 5 5	7,17	8,78
Fonemas	Ruso PCA30+SDC 1 5 5	7,11	
Fonemas	Ruso PCA30+SDC 2 5 4	6,64	6,61

relación con el sistema de base. Por una parte, se presentan los resultados relacionados con fonemas (Tabla 5.3) y por otra los obtenidos al usar estados fonéticos (Tabla 5.4). En ambas Tablas se presentan las configuraciones de base seguidas de las configuraciones óptimas en la que se indican los porcentajes de mejora obtenidos.

La Tabla 5.3 muestra los resultados relacionados con el sistema de SDPCs basados en fonemas y la Tabla 5.4 los de estados. Se muestran las métricas C_{avg} y se puede observar que los óptimos no son homogéneos, en especial en cuanto a la combinación de los parámetros M , dimensión de PCA, y K , el número de saltos considerados. Claramente en cada caso o sistema considerado va a hacer falta una experimentación y el hecho

CUADRO 5.4: C_{avg} para cada una de las configuraciones - Caso estados

	Sistema	C_{avg}	Mejora %
Estados	Húngaro PCA35+SDC 1 5 3	6,57	
Estados	Húngaro PCA70+SDC 2 4 3	6,44	1,98
Estados	Checo PCA35+SDC 1 5 3	7,33	
Estados	Checo PCA35+SDC 2 5 2	6,98	4,77
Estados	Ruso PCA35+SDC 1 5 3	7,00	
Estados	Ruso PCA50+SDC 2 5 5	6,35	9,28

CUADRO 5.5: Fusión del sistema SDPCs con el sistema acústico basado en MFCCs.

Sistema		C_{avg}	Mejora %
Acústico		7,60	
Fonemas	SDPCs	3,89	
Fonemas	SDPCs+Acústico	3,28	15,68
Estados	SDPCs	3,60	
Estados	SDPCs+Acústico	3,19	17,19

de que la dimensión final del vector sea $M(K + 1)$ va a ser clave en el óptimo que se encuentre. Sí que se ha comprobado que los óptimos se encuentran siempre para valores del producto $M(K + 1)$ similares.

Con la fusión de los mejores sistemas de cada HMM por fonema y por estado se llegó a obtener un C_{avg} de 3,89 que supone una mejora de hasta un 12,39 % en relación al 4,44 conseguido con la fusión de los sistemas de cada HMM sin el proceso de optimización seguido aquí (D'Haro y col., 2014).

Por último, se fusionó el sistema con los SDPCs optimizados con el acústico basado en MFCCs (ver Capítulo 3) obteniendo los resultados que se muestran en la Tabla 5.5. De nuevo, podemos observar cómo dicha fusión ofrece mejoras significativas, hasta del 17 %.

5.2.6. Conclusiones y aportaciones

El uso de los PLLRs a nivel de estado ha mejorado la tasa de reconocimiento al compararlo con el sistema a nivel de fonema. Fue necesario aplicar reducción de dimensionalidad dado el incremento de la dimensión del vector de parámetros. Con los PLLRs se han formado los SDPCs que incluyen información de contexto temporal gracias a la aplicación y optimización de la técnica SDC en los PLLRs y se ha mejorado la tarea LID, aunque se pudo notar que el sistema es sensible a las variaciones de los parámetros SDC, en especial con M (dimensión del PCA) y K (número de saltos de contexto considerados). Los óptimos van a depender de la tarea aunque se suelen obtener para un mismo valor aproximado del producto $M(k+1)$.

La aportación estuvo en el estudio de los PLLRs a nivel de estado fonético, el tratamiento de su contexto temporal y su optimización, además de la definición de una configuración estándar para el control del contexto temporal.

Capítulo 6

APLICACIÓN DE REDES NEURONALES RECURRENTES A SISTEMAS LID

6.1. Descripción de la técnica usada para incorporar ngramas-fonéticos en las RNNs

Vamos a describir el enfoque que se ha seguido para utilizar Redes Neuronales Recurrentes (RNNs) con ngramas-fonéticos como unidades de entrada para generar los modelos de lenguaje y distintas alternativas que hemos abordado para aumentar su robustez y mejorar la tarea de reconocimiento.

En su estructura, una RNN cuenta con tres capas, una de entrada $x(t)$, una de estado $s(t)$ y una de salida $y(t)$. La capa de entrada $x(t)$ (Figura 6.1) contiene un número de neuronas igual al número de ngramas-fonéticos existentes en el vocabulario. Para calcular la probabilidad de ocurrencia del evento futuro en la capa de salida $y(t)$ cuyo tamaño también es igual al número de ngramas-fonéticos. Así, se toma el ngrama-fonético presente en la secuencia y se activa su respectiva neurona en la capa de entrada $x(t)$ dejando inactivas las demás. Para ello utilizamos la codificación 1- N (donde N es el tamaño del vocabulario).

A partir de la suma ponderada de todas las combinaciones de pesos generados en la capa de entrada a partir de la neurona activada se obtienen los valores de las neuronas de la capa de estado $s(t)$, a los que se les aplica una función no-lineal (normalmente una sigmoide) para posteriormente avanzar por medio de la matriz de proyección W a la capa de salida $y(t)$ que tiene igual tamaño que la de entrada. A cada neurona de esta capa se le aplica la función Soft-max con lo que los valores de salida adquieren las características de probabilidades condicionales y se pueden utilizar para el cálculo de los modelos de lenguaje.

El procedimiento descrito, también llamado *Feed-Forward*, se implementa en el proceso de aprendizaje, que permite estimar los pesos que utiliza la RNN en la fase de evaluación. Los modelos de lenguaje generados aportan poco en la caracterización de un idioma si no se utiliza información de contexto en el proceso. Es en este punto, donde el concepto de *recurrencia*, que se define como la conexión directa entre la capa de estado $s(t)$ y la capa de entrada $x(t)$, introduce información del pasado en la red, mejorando la creación de los modelos de lenguaje. La información histórica o pasada se recupera en la fase de aprendizaje (algoritmo de retro-propagación en el tiempo) del tiempo $(t-1)$ ó incluso de tiempos anteriores representados en $s(t-1)$, siendo ésta usada junto con la información de la capa de entrada $x(t)$ en el tiempo (t) para generar los valores en la capa de estado $s(t)$ a los que se les aplica la función no-lineal descrita en el párrafo anterior.

Los modelos de lenguaje basados en RNNs se obtienen en la capa de salida $y(t)$ al multiplicar las probabilidades condicionales de los ngramas-fonéticos más probables en la capa de salida $y(t)$ de la red, para cada uno de los ngramas-fonéticos presentes en la entrada. En la Figura 6.1, se muestra el procedimiento descrito para el caso del n-gram fonético i_J_o .

El valor obtenido en la capa de salida y más específicamente, en la salida que se corresponde con el ngrama-fonético activado en la entrada, se utiliza con dos fines. El primero es para la fase de aprendizaje, donde se le compara con el valor esperado (que normalmente es el mismo que el de la activación) es decir 1 y el resultado de esta comparación es el valor de error que se retro-propaga. El segundo es la obtención de un modelo de lenguaje que se define cuando la red ya ha ajustado los pesos en la capa de entrada. En este caso, las salidas de la red, al ser el resultado de un proceso no lineal (óptimo para un proceso de clasificación) en la capa oculta y de un posterior proceso de normalización (función Soft-max) aplicado en la capa de salida representan una distribución de valores donde la salida que se corresponde con el ngrama-fonético presente en la entrada se considera como la probabilidad de ocurrencia del siguiente ngrama-fonético y pasa a formar parte del cálculo del modelo de lenguaje.

Dada esta correspondencia entre el ngrama-fonético presente en la capa de entrada y la probabilidad de ocurrencia definida en la capa de salida, se requiere que la capa de salida tenga el mismo tamaño que la capa de entrada, es decir el tamaño del vocabulario N . La repetición de este proceso para todo el conjunto de secuencias de ngramas-fonéticos posibilita la generación de un modelo de lenguaje.

Hemos utilizado el criterio de simplificación propuesto por Mikolov (Mikolov, 2012) con el que se evita la duplicación del tamaño de la capa de entrada (Bengio y col., 2006) para dar cabida al nuevo ngrama-fonético en la codificación $1-N$.

FIGURA 6.1: Arquitectura típica de una Red Neuronal Recurrente

FIGURA 6.2: Sistema de reconocimiento de idioma

La incorporación de información de ngramas-fonéticos anteriores se realiza mediante algoritmos de aprendizaje que incluyen la variable del tiempo, como por ejemplo, el BPTT (Werbos, 1990) o el RTRL (Chang, Chang y Huang, 2002). En nuestro caso, dado su amplia utilización para la generación de modelos de lenguaje (Liu y col., 2014), hemos elegido el algoritmo de retro-propagación en el tiempo BPTT (Werbos, 1990), (Guo, 2013).

De lo expuesto anteriormente se puede observar que las RNNs funcionan igual que las NNs en la fase de definición de salidas *Feed-Forward* y que la diferencia radica en la fase de aprendizaje *Back-Propagation* de la red, en la que se definen los pesos con los que se ajustan las entradas para provocar las salidas deseadas. Es en esta fase donde la red adquiere la capacidad de *aprovechar* información de instantes anteriores y utilizar esa información para añadirla a las nuevas entradas. En este trabajo hemos utilizado como base el software de rnnlm.org en razón de que es de libre distribución y de que se adecua a nuestras necesidades.

6.2. Estructura de la experimentación

El sistema utilizado para evaluar el uso de modelos de lenguaje basados en RNN y n-gramas fonéticos (que hemos bautizado como RNNLM-P) en tareas LID está formado por dos componentes: Front-End y Back-End. En el Front-End se obtienen los vectores de características que se utilizan para el entrenamiento de la red, en tanto que la definición, uso de los modelos para el reconocimiento y la toma de la decisión final se lleva a cabo en el componente Back-End, tal y como se muestra en la Figura 6.2.

El componente Front-End se compone de la fase de pre-procesamiento de la señal de voz y se completa con un reconocedor que es el que permite generar las secuencias fonéticas que se corresponden con las señales de audio de la entrada. Los reconocedores de fonemas pueden ser de varios tipos, siendo los más comunes aquellos que utilizan

HMMs para definir la probabilidad de una secuencia de características observadas en el evento acústico $O = o_1, \dots, o_T$ dado un modelo M y la secuencia de estados $Q = q_1, \dots, q_T$ que mejor explican las características observadas. Por otro lado están los sistemas híbridos ($NN+HMM$) (que es el que hemos usado) en los cuales se entrenan NNs para proyectar los parámetros de entrada en probabilidades a posteriori de cada fonema a la salida de la red, sumando con ello los efectos discriminativos de las redes neuronales artificiales en los parámetros de entrada. Así, hemos utilizado el reconocedor de fonemas de la Universidad de Brno (Schwarz, 2009) con el que se obtienen 3 conjuntos de secuencias de fonemas, uno por cada idioma que ofrece dicho reconocedor.

En la fase de entrenamiento de la red, se han agrupado los ficheros de secuencias de fonemas en función de su idioma y del HMM con el que han sido generados. El tipo de entrenamiento utilizado para los RNNLM-P es supervisado y el resultado de la fase de entrenamiento es la creación de los modelos de lenguaje RNNLM-P, uno para cada modelo de reconocimiento HMM y para cada idioma de entrada. Dado que trabajamos con la base de datos KALAKA-3 en su condición *plenty* y *closed*, al ser tres los modelos HMM y seis los idiomas de entrada, generamos en total 18 modelos de lenguaje.

El Back-End de nuestro sistema LID se caracteriza por ser el componente en el que se comparan las secuencias fonéticas de un idioma desconocido con los modelos de idioma creados. Para ello hemos utilizado una estructura similar a la de PPRLM propuesta por (Zissman, 1996) para tareas LID. De esta comparación se obtienen *scores*, que se definen como distribuciones de probabilidad de que el idioma de la señal evaluada se corresponda con el idioma de uno de los modelos entrenados. Estos *scores* se utilizan como entradas de un clasificador donde (dependiendo de las condiciones de evaluación) se elige el idioma con el que se maximice el score. Los modelos se generan mediante la aplicación de diferentes técnicas. Entre algunas de ellas cabe mencionar: las redes neuronales (Bengio y col., 2006) y las técnicas de suavizado, como las propuestas por Good-Turing, Jelinek-Mercer, Witten-Bell, Ney y otros. En nuestro caso hemos utilizado RNNs (Mikolov y col., 2011) y los modelos de Witten-Bell (Witten y Bell, 1991) y Kneser-Ney (Goodman, 2001b).

Complementando la etapa de Back-End está el clasificador, encargado de definir el idioma al que pertenece la señal de voz de evaluación. Existen varios tipos de clasificadores, entre los más conocidos están los de regresión logística multi-clase (MLR) (Brummer y Leeuwen, 2006) y los que utilizan SVM (Weston y Watkins, 1999).

Es importante indicar que para todas las secuencias de fonemas, tanto las utilizadas para el entrenamiento como para el reconocimiento, se ha suprimido la información referida a: silencios (pau), ruidos del locutor (spk) y ruidos no estacionarios no relacionados con el locutor (int).

FIGURA 6.3: Sistema LID basado en RNNLMs

La métrica de evaluación que se ha utilizado es la función de coste de detección C_{avg} que pondera los errores de falsa aceptación (e.g. reconocer un determinado fichero con un idioma distinto al que realmente es) y los de falso rechazo (e.g. no reconocer el idioma real de un determinado fichero) (Martin y Greenberg, 2010) en su cálculo, véase ecuación 3.2.

Con el fin de aprovechar la información generada por los tres reconocedores de fonemas, para los resultados finales hemos fusionado los modelos de Ruso, Húngaro y Checo del reconocedor de fonemas. Finalmente, incluimos en la fusión al sistema acústico de referencia descrito en el Capítulo 3. En la Figura 6.3 se muestra el diagrama completo sobre el cual se llevó a cabo la experimentación.

6.2.1. Descripción de los parámetros configurables de la RNN

Para conseguir unos resultados óptimos con RNNLMs-P es imprescindible definir una configuración adecuada de sus parámetros. Entre los más importantes podemos mencionar los siguientes:

Número de neuronas en la capa de estado (NNE)

Este parámetro depende, por una parte de la cantidad de información utilizada para el entrenamiento de los modelos y, por otra, del tamaño del vocabulario. El objetivo es

que el NNE sea lo suficientemente grande para que la red tenga un número suficiente de pesos estimados que garanticen su buen entrenamiento. Sin embargo, debe evitarse tener un valor excesivo, dado que el coste computacional crece de manera importante junto con la posibilidad de sobre-entrenar la red (Mikolov, 2012).

Número de clases (NCS)

Con el fin de acelerar el proceso de entrenamiento de la red se ha experimentado con varias propuestas, que van desde la disminución del número de neuronas en la capa de estado o del número de iteraciones realizadas para la localización de un mínimo, hasta la factorización de la capa de salida (Goodman, 2001a), (McClelland, Rumelhart y Pdp, 1986). En este contexto, la factorización de la capa de salida tiene muy buena aplicabilidad, como se muestra en (Mikolov, 2012), aunque como en todos los casos, afecta a la precisión de la predicción. La idea consiste en calcular la probabilidad de una clase dada la historia y luego la del ngrama-fonético dada la clase, siendo la probabilidad del ngrama-fonético dada la historia el producto de las dos (Mikolov, 2012), (McClelland, Rumelhart y Pdp, 1986), (Goodman, 2001a).

Al factorizar la capa de salida se asume que la probabilidad $P(\omega | \omega_1, \dots, \omega_{(i-1)})$ de un ngrama-fonético ω dada su historia reciente i es igual al producto de la probabilidad de la clase $P(\text{clase}(\omega) | \omega_1, \dots, \omega_{(i-1)})$ a la que pertenece el ngrama-fonético ω dada su historia reciente i y de la probabilidad $P(\omega | \omega_1, \dots, \omega_{(i-1)}, \text{clase}(\omega))$ del ngrama-fonético dada su historia reciente i y su clase. Las dos probabilidades factorizadas se calculan de forma separada, pudiendo primero determinar la probabilidad de la clase a la cual pertenece el ngrama-fonético ω y luego calcular la probabilidad del ngrama-fonético ω pero solo para los ngramas-fonéticos que están dentro de esa clase predicha en el paso anterior (Chen y col., 2015).

La red que se muestra en la Figura 6.4 genera, por una parte, una salida W que define la probabilidad de que un ngrama-fonético ω_i se corresponda con la historia $P(\omega_i | s_i)$, y por otra, genera una salida C que define la probabilidad de que ese ngrama-fonético pertenezca a una clase en particular $P(C_i | \omega_i)$ (Mikolov y col., 2010). *Se consigue acelerar el proceso cuando se ubica el ngrama-fonético dentro de una clase (subconjunto del conjunto de datos) en vez de ubicarlo dentro de todo el conjunto de datos.*

Por ello, en un primer paso se define la clase en la cual se ubicaría y luego el ngrama-fonético dentro de esa clase. Cuantas más clases se definan, más rápido se encuentra el ngrama-fonético, pero afecta en la precisión ya que al aumentar el número de clases también aumenta la probabilidad de que el ngrama-fonético no aparezca en la clase elegida, y por tal, la respuesta pierde exactitud. El producto de las dos probabilidades es

FIGURA 6.4: Factorización de la capa de salida

la distribución de probabilidad de que el ngrama-fonético presentado en la entrada se corresponda con la salida activada.

Así, en el contexto de los modelos estadísticos de lenguaje, la distribución de probabilidad de una secuencia de ngramas-fonéticos $P(\omega_1 \omega_2 \dots \omega_n)$ puede ser factorizada en un producto de estimación de probabilidades de todos los ngramas-fonéticos condicionados a su historia, tal como se muestra en la ecuación 6.1:

$$P(\omega_1 \omega_2 \dots \omega_n) = \prod_{1 \leq i \leq n} P(\omega_i | h_i) \quad (6.1)$$

Donde n es el número total de ngramas-fonéticos y,

$$h_i = \omega_1 \omega_2 \dots \omega_{i-1}, \text{ es la historia previa de cada fonema} \quad (6.2)$$

Cada probabilidad factorizada $P(\omega|h)$ se genera a partir de dos elementos, el primero a partir del ngrama-fonético previo ω_{i-1} que se presenta a la entrada de la red y para el cual se aplica una codificación 1-N y el segundo es la información codificada del vector de estado del tiempo inmediatamente anterior $s_{(i-1)}$, por lo que:

$$P(\omega_i | s_i) = P(\omega_i | \omega_{i-1}, s_{i-1}) = P(\omega_i | h_i) \quad (6.3)$$

La probabilidad a posteriori del ngrama-fonético predicho se obtiene aplicando la función de activación Soft-max en la capa de neuronas de salida de la red en la salida correspondiente.

Por otro lado, asumiendo que los ngramas-fonéticos en la capa de salida pueden ser factorizados en *clases*, a partir de un ngrama-fonético en una de las entradas de la red se puede estimar la probabilidad a posteriori de que éste corresponda a una de las clases

en la capa de salida y luego estimar la probabilidad de que el ngrama-fonético dentro de dicha clase aparezca dada la historia previa de aparición de otros ngramas-fonéticos. Con ello, la probabilidad a posteriori de un ngrama-fonético se define como el producto de las dos distribuciones de probabilidad estimadas, la una sobre las clases y la otra del ngrama-fonético dentro de una clase, así:

$$P(\omega_i|s_i) = P(C_i|\omega_i) * P(\omega_i|s_i) \quad (6.4)$$

Para el caso de que no se factorice la capa de salida de la RNN se asume que la red posee una sola clase, con lo que la probabilidad de que el ngrama-fonético se corresponda con esa clase es igual a 1. Un valor alto del número de clases (NCS) acelera el entrenamiento de la RNN pero el modelo de lenguaje final va a resultar menos preciso. Las clases se forman automáticamente utilizando las distancias basadas en un árbol de Huffman.

Memoria de la RNN (MEM)

Con este parámetro se introduce en la red neuronal información del pasado (histórica) para ampliar el contexto temporal del modelo obtenido. Por ejemplo, en la Figura 6.5 puede observarse la estructura de una red neuronal recurrente que utiliza información de contexto contenida en la capa de estado en los 3 instantes anteriores $s(t-1)$, $s(t-2)$ y $s(t-3)$, utilizando un valor de MEM igual a 3. La misma figura es la base del algoritmo de retro-propagación en el tiempo (BPTT).

6.2.2. Configuración de la red neuronal

Vamos a considerar como base o de referencia un experimento en el que como entrada de la red se utilizan las secuencias de fonemas sin considerar n-gramas, de manera similar a lo propuesto por Mikolov (Mikolov, 2012) y (Mikolov y col., 2011), y utilizar los parámetros óptimos de la red neuronal como punto de partida.

Dichos parámetros corresponden a una red neuronal sin factorización en la capa de salida (NCS=1) donde el vocabulario de fonemas es pequeño y el coste computacional es bajo.

Para hallar una primera aproximación a los parámetros óptimos, utilizamos como métrica de comparación entre sistemas a la entropía promedio obtenida para los ficheros de evaluación y cada uno de los modelos de idioma obtenidos en la fase de entrenamiento de la red. La razón es el menor coste para obtenerla.

FIGURA 6.5: MEM en Redes Neuronales Recurrentes

El primer parámetro a definirse ha sido el número óptimo de neuronas en la capa de estado NNE y para ello, hemos partido de NCS=1 y MEM=10. En la Figura 6.6 se muestra el comportamiento de la red en estas condiciones:

En la Figura 6.6 se puede ver cómo la red presenta una zona estable en torno a 80, por lo que lo hemos elegido como el valor óptimo (NNE=80).

Tras haber determinado el valor óptimo para NNE realizamos los experimentos correspondientes para obtener el óptimo de MEM que es la cantidad de información histórica que la red integra en el cálculo del modelo de lenguaje. Para ello, fijamos los primeros parámetros NCS=1 y NNE=80, variando el valor de MEM hasta encontrar su óptimo.

FIGURA 6.6: Efectos en la perplejidad para el conjunto de evaluación al modificar NNE

FIGURA 6.7: Resultados al modificar MEM

CUADRO 6.1: Parámetros óptimos de la RNN usando 1-grams en la base de datos KALAKA-3.

	NNE	MEM	NCS
1grams	80	3	1

En la Figura 6.7 se puede observar que a partir de un valor MEM=3 la mejora empieza a saturarse y, por lo que se puede afirmar que no compensa utilizar valores elevados de MEM para una mejora tan reducida, ya que como se recomienda en (Zaremba, Sutskever y Vinyals, 2014) siempre funciona mejor un RNNLM de tamaño reducido. Por lo tanto, hemos utilizado MEM=3 como parámetro de partida.

En resumen, se pueden observar en la (Tabla 6.1) los parámetros que se han utilizado como base en la experimentación.

6.3. Efectos del uso de ngramas-fonéticos en la generación de RNNLMs

La configuración óptima de los RNNLMs generados con ngramas-fonéticos (Ngramas) depende del tamaño del inventario de unidades fonéticas utilizado. Es evidente que se produce un incremento significativo del mismo al usar 2 ó 3gramas y la codificación 1-N. En el caso de 2gramas el inventario de unidades fonéticas El tamaño del vocabulario crece hasta en 30 veces y hasta 500 veces al usar 3gramas, ambas en comparación con el inventario de 1gramas. En la Tabla 6.2 se muestra a modo de ejemplo, la relación existente entre 1-, 2-, y 3gramas-fonéticos y el correspondiente inventario generado para el caso de datos de entrenamiento de español y para cada uno de los modelos del reconocedor fonético.

CUADRO 6.2: Número de ngramas-fonéticos por cada modelo del reconocedor fonético.

ngramas-fonéticos	Ruso	Húngaro	Checo
1grams	52	61	46
2grams	1876	1938	1572
3grams	29822	28097	25874

6.3.1. Número de iteraciones al aumentar el orden del n-grama

El incremento del inventario de ngramas fonéticos de orden superior dificulta el entrenamiento de la RNN debido a la dispersión de información, provocada por la aparición de nuevas unidades fonéticas de muy poca frecuencia de ocurrencia, lo que incrementa en la etapa de reconocimiento el número de ngramas fonéticos no vistos en la fase de entrenamiento. En la creación de los modelos de lenguaje, la probabilidad condicional asignada por la red neuronal a los *ngramas-fonéticos* no vistos en el entrenamiento es muy baja, lo que provoca una fuerte penalización en el valor de la entropía proporcionado por la red, lo que lleva a un empeoramiento del rendimiento del sistema.

De un primer análisis hemos podido comprobar que para 2grams, si bien, existe un incremento del inventario, este no repercute significativamente en el rendimiento de la red. Sin embargo, para 3grams la situación resulta diferente y por ello, aplicamos un valor de probabilidad mínima (PM) a dichos ngramas-fonéticos con el fin de reducir este problema. Veremos en la sección 6.3.4 un análisis detallado de los resultados obtenidos para 3grams-fonéticos.

Por otro lado, el incremento del inventario implica también la creación de modelos de lenguaje más grandes. Esta situación no sería negativa si los ngramas-fonéticos de una secuencia fonética y, en general, de la base de datos, aparecieran de manera más o menos similar, pero en nuestro caso, como hemos dicho, el incremento del inventario (particularmente para el caso de 3grams o unidades de mayor orden) genera un número considerable de ngramas-fonéticos con un número muy reducido de apariciones en la fase de entrenamiento. Esto provoca que el modelo sea más grande lo que dificulta una estimación fiable de los pesos de la red.

Una de las estrategias utilizadas para limitar el tamaño de los modelos ha sido controlar la mejora mínima que se exige a la red neuronal para detener el proceso de entrenamiento. De esta manera la red queda entrenada mediante un número mayor o menor de iteraciones, variando con ello, el tamaño del modelo. Tomando en cuenta lo descrito en (Zaremba, Sutskever y Vinyals, 2014), un modelo relativamente pequeño, tiende a ser más generalista y por ende más robusto.

CUADRO 6.3: Valores de mejora mínima en la generación de modelos de lenguaje.

	1grams	2grams	3grams
Iteraciones promedio	80	40	10

En una experimentación previa comprobamos que se podía utilizar el mismo valor de mejora mínima en la red neuronal tanto para 1gramas como para 2gramas, lo cual no sucede en el caso de 3gramas, en el que se ha limitado el número de iteraciones por medio de un valor de mejora mínima más alto para reducir, de este modo, la posibilidad de un sobre-entrenamiento. En resumen, los modelos generados para 1-, 2- y 3gramas-fonéticos han sido generados con los valores de mejoras mínimas que se muestran en la Tabla 6.3.

Con todas estas consideraciones, tomamos como referencia los parámetros de base ya definidos, es decir NNE=80 y MEM=3, excepto para el número de clases en la capa de salida que lo hemos definido como NCS=30 debido al tiempo excesivo que tarda el entrenamiento con un NCS menor y las desventajas de tener un modelo muy grande.

6.3.2. Para 1gramas

Para el caso de fonemas como unidades fonéticas se ha establecido NCS=1 porque debido al tamaño del vocabulario (véase Tabla 6.2) no es necesario factorizar la capa de salida. Respecto a la cantidad de información histórica que la red neuronal utiliza para su entrenamiento, hemos definido MEM=3 como parámetro de partida. Para definir el número óptimo de neuronas en la capa de estado, se han realizado varios experimentos y al observar que a partir de 250 las mejoras no resultaron significativas, definimos el número óptimo de neuronas como dicho valor. En la Figura 6.8, se muestra el comportamiento de C_{avg} para cada uno de los HMMs del reconocedor fonético al utilizar 1gramas como unidades fonéticas. Como se puede ver, el HMM de Ruso es el que mejor resultado entrega en comparación los modelos de Húngaro o Checo.

Habiendo definido el número óptimo de neuronas en la capa de estado, el último parámetro por definir ha sido el valor de MEM, mediante el cual la red neuronal recupera información del pasado para incluirla en la generación del modelo de lenguaje. Considerando a C_{avg} como la métrica de evaluación, se han utilizado los parámetros ya optimizados (NCS, y NNE) variando el parámetro MEM entre 1 y 10. En estas condiciones, obtuvimos una mejora en la tarea de reconocimiento al procesar fonemas o 1gramas

FIGURA 6.8: Cavg de cada uno de los HMMs utilizando 1gramas

FIGURA 6.9: Cavg con parámetros óptimos para 1gramas

con NNE=250 y MEM=3. Así, la red quedó definida con NNE=250, MEM=3 y NCS=1 (Tabla 6.4).

Como información adicional, en la Figura 6.9 se muestra el C_{avg} utilizando NNE=60 y NNE=250, pudiéndose observar un mejor rendimiento en este último caso.

6.3.3. Para 2gramas-fonéticos

Como ya hemos comentado, el tamaño del vocabulario se incrementa al considerar 2gramas-fonéticos lo que implica modificar los parámetros óptimos. En el caso de 2gramas se llega a entrenar la red con un vocabulario de entre 1,5 y 2k ngramas-fonéticos (Tabla 6.2), valor que depende de la cantidad de fonemas existentes en cada uno de los idiomas del reconocedor (Schwarz, 2009). Al tener más de 1k ngramas-fonéticos en la entrada y salida de la red, hemos considerado conveniente llevar a cabo una factorización de la capa de salida que permita acelerar su entrenamiento pero cuidando de no disminuir significativamente la precisión en la tarea de reconocimiento. Para definir el

FIGURA 6.10: Resultados variando la factorización de la capa de salida para 2gramas-fonéticos en la generación de RNNLMs

NCS óptimo utilizamos el C_{avg} de evaluación, observando el comportamiento de la red al variar el número de neuronas en la capa de estado (NNE) para 3 casos de factorización de la capa de salida (NCS), 1, 30 y 60. Para todos los experimentos se utiliza la fusión de los tres modelos del reconocedor de fonemas.

En la Figura 6.10 se puede ver el comportamiento de 2gramas para tres casos de factorización de la capa de salida. En términos generales, se puede observar una tendencia de mejora en la tarea de reconocimiento a medida que aumenta el número de neuronas en la capa de estado, hasta llegar a un límite en el que se sobre-entrena.

- Sin factorizar la capa de salida, es decir NCS=1, la tarea de reconocimiento funciona de una manera inestable, mejorando y empeorando para diferentes valores de NNE. De todos los experimentos, al hablar de tendencia es el que menos mejora con el aumento de NNE. Entendemos que la situación se produce debido al sobre-entrenamiento de la red y que por ello, la tarea de reconocimiento se ve perjudicada. Para este caso, en ningún momento los resultados llegan a ser mejores que los obtenidos por 1gramas.

- Factorizando la capa de salida con NCS=60, se nota de una forma más clara la tendencia a mejorar la tarea de reconocimiento conforme se aumenta el valor de NNE, obteniendo un límite en NNE=80. Si bien la tendencia de mejora es mayor conforme aumenta NNE, al igual que con NCS=1 el sistema no mejora a lo obtenido con 1gramas.

- Al utilizar el valor intermedio, es decir, al factorizar la capa de salida con NCS=30 se logra una tendencia de mejora más significativa que en los dos casos anteriores, llegando a mejorar el rendimiento obtenido utilizando 1gramas. El valor de mejora se puede ver en la Tabla 6.4.

En general, hemos podido encontrar que en el punto común de NNE=100 aparece el óptimo tanto de 1gramas como de 2gramas, siendo ligeramente mejor el de 2gramas

FIGURA 6.11: Cavg óptimos de 1gramas y 2gramas con MEM variable

CUADRO 6.4: Comportamiento del Cavg para 1gramas y 2gramas.

Experimento LID	C_{avg}	Mejora %
1gramas	12,81	
2gramas-fonéticos	12,40	3,2

pero sugiriendo que la concatenación de fonemas de una frase utilizados como unidades fonéticas puede mejorar el rendimiento de sistemas de reconocimiento de idioma a nivel fonotáctico, como es nuestra hipótesis.

Habiendo definido el parámetro $NCS=30$, nos queda realizar un barrido del parámetro MEM con el fin de encontrar su valor óptimo, ya que, aunque se haya utilizado un valor de MEM para la definición del NCS óptimo, éste se correspondía con los parámetros de base propuestos inicialmente y no a los valores óptimos de 2gramas-fonéticos. Con el barrido de MEM hemos obtenido los resultados que se pueden ver en la Figura 6.11 y que muestran los óptimos de 1gramas y 2gramas.

Si bien la Figura 6.11 muestra mejoras continuas de C_{avg} conforme el valor de MEM aumenta, a partir de $MEM=8$ las mejoras son mínimas. De la experimentación previa se pudo observar que si bien el uso de $MEM=20$ no ralentizaba el entrenamiento de la red tampoco generaba mejoras significativas respecto al valor de C_{avg} obtenido con $MEM=8$. Por ello, la configuración óptima para la generación de RNNLMs al usar 2gramas-fonéticos se estableció en $NCS=30$, $NNE=100$ y $MEM=8$.

6.3.4. Para 3gramas-fonéticos

Pasamos a estudiar los 3gramas-fonéticos. Como se ha dicho anteriormente, lo que se busca es aumentar la co-ocurrencia de eventos fonéticos y así caracterizar de mejor manera el idioma que representan.

FIGURA 6.12: Óptimo de Cavg con 3gramas-fonéticos variando MEM

Considerando que el tamaño de vocabulario generado está entre los 25k y 30 k-gramas-fonéticos (Véase Tabla 6.2), puede notarse un aumento de alrededor de 20 veces el número de unidades obtenidas para 2gramas-fonéticos. Este aumento drástico del vocabulario llevó al consecuente replanteamiento de los parámetros de la red neuronal en búsqueda de sus nuevos valores óptimos.

Los valores de partida de configuración de la RNN fueron los siguientes: NCS=100 y MEM=3. Si bien se partía de un número de neuronas en la capa de estado NNE=100, lo lógico al tener un mayor número de entradas en la red neuronal era que el NNE también aumentara. Teniendo esto en cuenta, obtuvimos un óptimo para 200 neuronas. Así mismo, la factorización óptima de la capa de salida también cambia obteniéndose un óptimo para NCS=300, que implica un tiempo prudencial para el entrenamiento de la red y además nos permite mejorar los resultados obtenidos con 2gramas. Ya por último, habiendo definido NNE y NCS para 3gramas, se debe determinar el MEM óptimo para dicho número de neuronas. Los resultados conseguidos se ven en la Figura 6.12.

Como se puede observar en la Figura 6.12, el aumento de la memoria utilizada por la RNN para recuperar información del pasado y utilizarla en el tiempo presente favorece la tarea de reconocimiento hasta un punto en el cual comienza a empeorar de forma paulatina y constante. Las mejoras respecto al sistema de base se han conseguido para configuraciones de MEM=2 y MEM=3.

Como explicación de por qué se produce este efecto de empeoramiento para MEM>3, entendemos que la dispersión intra y entre clases de los 3gramas-fonéticos provocada por el gran número de elementos poco frecuentes en las secuencias (mucho mayor que en el caso de 1gramas y 2gramas) genera un modelo con poca capacidad de relacionar los 3gramas presentados en la entrada de la red dentro de un contexto, ya que dichos 3gramas en su mayoría tienen pocas repeticiones para tener una estimación correcta. En la fase de evaluación es incluso peor, dado que los 3gramas-fonéticos que no aparecen en la fase de entrenamiento penalizan el rendimiento del modelo de lenguaje y se generan

CUADRO 6.5: log-probabilidades asignadas por las RNNLMs y por SRI-LM para dos 3-gramas no aparecidos en el entrenamiento.

3grama-fonético	RNNLM	SRI-LM
i-i:-x	-4,7917	-2,1383
m-i-u:	-4,6339	-2,2294

valores de entropía excesivos. Para corregir este efecto, hemos propuesto disminuir la penalización implícita que aplica la RNN en aquellos 3gramas que aparecen en la fase de evaluación pero que no aparecen en el entrenamiento de la red. La penalización se refiere a un empeoramiento drástico del *score* para unidades no vistas en el entrenamiento.

Para comprobar este fenómeno, comparamos las log-probabilidades obtenidas por los RNNLMs y las obtenidas con un modelo de lenguaje clásico (SRI-LM) (Stolcke, 2002). En la Tabla 6.5 se muestran los valores para dos 3gramas-fonéticos que no habían aparecido en la etapa de entrenamiento.

Como se puede observar en la Tabla 6.5 la penalización aplicada por las RNNs es mayor a la aplicada por los modelos de lenguaje clásicos (Chen y Goodman, 1996), y por ello, y como se mencionó anteriormente, hemos propuesto aplicar un valor de probabilidad mínima (PM) en el cálculo de la log-probabilidad de ocurrencia de cada uno de los ngramas-fonéticos de una secuencia. Con este valor buscamos normalizar los valores de probabilidad generados por cada ngrama-fonético, disminuyendo el efecto negativo provocado por la penalización aplicada por la RNN a los 3gramas que no habían aparecido en el entrenamiento.

El valor de probabilidad mínima aplicado se obtiene en dos pasos. En el primero calculamos el promedio de la probabilidad de ocurrencia de cada uno de los 3gramas de la secuencia evaluada y en el segundo dividimos este valor por una constante numérica k (en adelante, umbral) cuyo valor se elige de forma empírica buscando el mejor rendimiento de la red. El valor de PM aplicado a cada uno de los ficheros de evaluación se obtiene de acuerdo a la ecuación 6.5:

$$PM = \frac{1}{Nk} \sum_{1 \leq i \leq N} p(\omega i_t | \omega j_{t-1}) \quad (6.5)$$

Donde:

$p(\omega i_t | \omega j_{t-1})$ es la probabilidad de ocurrencia de un 3grama-fonético ω dado el aparecido en el tiempo anterior ωj_{t-1} .

k es la constante numérica

y;

N es el número de 3gramas en el fichero.

FIGURA 6.13: Resultados de modificar k en la aplicación del umbral

FIGURA 6.14: Comparación del Cavg aplicando o no el umbral

Con el fin de definir el valor óptimo del umbral k realizamos evaluaciones de su C_{avg} modificando k de 1 a 10 usando la configuración NNE=200, NCS=300 y MEM=8 donde hemos podido observar claramente el efecto del umbral en la disminución del efecto de la penalización aplicada por la RNN a los ngramas-fonéticos no vistos en el entrenamiento. Los resultados se pueden ver en la Figura 6.13.

En la Figura 6.13 se puede ver que se obtiene el mejor resultado para $k=5$ y por ello usamos esta constante para todos los experimentos posteriores. Habiendo definido k evaluamos el C_{avg} para los parámetros óptimos NNE=200, NCS=300 con MEM variable, así hemos podido observar el comportamiento del sistema en la tarea de reconocimiento. El resultado se muestra en la Figura 6.14, donde además de la gráfica que incluye el efecto del umbral, se ha incluido la curva de no aplicar el umbral (Figura 6.12) con el fin de comparar la mejora producida.

Tal como se puede ver en la Figura 6.14 se obtuvieron mejoras en la tarea de reconocimiento al aplicar dicho umbral. Encontramos un mínimo para MEM=5 con una mejora aproximada del 2 % relativo. Esto prueba que la normalización llevada a cabo mediante

FIGURA 6.15: Cavg óptimo de 1gramas, 2gramas y 3gramas-fonéticos

el valor de *probabilidad mínima* ha sido útil para disminuir la penalización aplicada por la RNN a los ngramas-fonéticos no vistos en el entrenamiento.

Luego usamos la configuración de 3gramas, NNE=200, NCS=300 y MEM=5 para fusionar los resultados previos encontrados con 1 y 2gramas en la tarea de reconocimiento.

También se ha aplicado el concepto de *probabilidad mínima* para el caso de 2gramas, pero las mejoras han resultado mínimas, probablemente porque la cantidad de ngramas-fonéticos no vistos en la fase de entrenamiento es muy pequeña en comparación con el total de ngramas-fonéticos vistos.

Finalmente, vamos a comparar en la misma figura los efectos producidos al utilizar ngramas-fonéticos de distinto orden en la generación de modelos de lenguaje con RNNs en tareas LID. Así, en la Figura 6.15 mostramos el comportamiento de C_{avg} al utilizar RNNLM-P en unidades fonéticas de uno, dos y tres elementos, donde comprobamos que al usar 3gramas-fonéticos conseguimos la mejor tasa de reconocimiento comparando lo obtenido al usar 1gramas que es la línea base de experimentación (7,7 % relativo en MEM=5) ó 2gramas (5 % relativo en MEM=8).

6.3.5. Fusión de los sistemas RNNLM-P, PPRLM Y Acústico

Como se ha dicho en los apartados previos, se van a fusionar los sistemas de 1-, 2 y 3gramas-fonéticos para obtener una única métrica relativa al uso de ngramas-fonéticos en tareas LID. Los resultados se muestran en la Tabla 6.6, donde se han considerado las configuraciones óptimas para cada caso:

Para 1gramas: NNE=250, MEM=3 y NCS=1.

Para 2gramas: NNE=100, MEM=8 y NCS=30.

Para 3gramas: NNE=200, MEM=5 y NCS=300.

CUADRO 6.6: Fusión de los ngramas-fonéticos de 1ero, 2do y 3er orden.

Experimentos RNNLM-P	C_{avg}	Mejora %
1grams	12,81	
2grams	12,09	5,6
3grams	12,02	6,2
Fusion	11,15	14,4

CUADRO 6.7: Comparación de RNNLM-P con PPRLM y Acústico.

Experimentos LID	C_{avg}	Mejora %
Acústico	7,60	
PPRLM	11,57	
RNNLM-P	11,15	3,6

Como se puede ver en la Tabla 6.6, la fusión de los tres tipos de ngramas-fonéticos favorece la tarea de tal manera que se llega a obtener hasta el doble de mejora respecto a lo conseguido por 2gramas o 3gramas por separado, además de que muestra ya un valor representativo para ser comparado con otras técnicas de reconocimiento o inclusive para ser utilizado como complemento para mejorar mediante fusión, sistemas de reconocimiento de idioma con otras fuentes de información, como por ejemplo, nuestro sistema acústico de referencia.

Precisamente, en la Tabla 6.7 se puede ver el C_{avg} obtenido por el sistema RNNLM-P propuesto en este trabajo y el C_{avg} obtenido, tanto, por el sistema PPRLM (modelo de lenguaje clásico), como por el sistema acústico basado en MFCCs, que son sistemas ya conocidos para tareas LID y que, como se podrá ver, mejoran con la fusión del sistema RNNLM-P. En el caso del sistema PPRLM se utilizaron los resultados obtenidos al generar los modelos de lenguaje con trigramas aplicando la técnica Witten-Bell (Witten y Bell, 1991) para el suavizamiento.

En la Tabla 6.8 se puede ver el resultado global de las fusiones en donde hemos combinado los sistemas RNNLM-P, PPRLM y Acústico de diversas formas, encontrando mejoras para todos los casos. Las principales mejoras se han presentado al fusionar el sistema RNNLM-P con el sistema acústico basado en MFCCs donde llega a conseguirse una mejora de hasta un 37 % relativo en el C_{avg} .

CUADRO 6.8: Fusiones de los sistemas PPRLM, Acústico y RNNLM-P.

Experimentos LID	C_{avg}	Mejora %
PPRLM	11,57	
RNNLM-P+PPRLM	10,74	7,2
Acústico	7,60	
PPRLM+Acústico	5,10	32,9
RNNLM-P+Acústico	5,06	33,4
RNNLM-P+PPRLM+Acústico	4,80	36,8

6.3.6. Conclusiones y aportaciones

Hemos adaptado las RNNs y sus modelos de lenguaje, que normalmente están diseñados para trabajar a un nivel léxico, a un contexto fonético. Para ello, hemos usado parámetros que tienen en cuenta la información de contexto en las secuencias de fonemas (ngramas fonéticos). Con ello las secuencias de fonemas se convirtieron en secuencias de unidades más extensas sin llegar a un nivel léxico. Con el concepto de probabilidad mínima hemos conseguido un 13 % de mejora en C_{avg} comparado con el sistema basado en fonemas y un 3,6 % en comparación con el sistema de referencia basado en PPRLM. En la fusión con el acústico de referencia obtuvimos una mejora de 36,8 %.

Las aportaciones de esta línea de investigación son: La adaptación de los RNNLMs (diseñado para palabras) al ámbito fonético para tareas de LID y la introducción del concepto de probabilidad mínima para disminuir la penalización de la RNN a los ngramas fonéticos no vistos en el entrenamiento.

Capítulo 7

APLICACIÓN DE EMBEDDINGS NEURONALES EN TAREAS LID

7.1. Creación de embeddings neuronales fonotácticos

En este capítulo vamos a describir el enfoque que se ha seguido para poder aplicar los modelos de representación vectorial (embeddings neuronales) a nivel fonético en tareas de reconocimiento de idioma.

Para ello, hemos usado la información contenida en la capa de estado de una red neuronal, que es de donde extraemos los vectores que identifican numéricamente a los ngramas-fonéticos presentes en la entrada de la red y son los que luego utilizamos para definir sus modelos (a los que llamaremos en adelante *embeddings neuronales* ó *ENs*). Por otra parte, en un ámbito más general como es el de los sistemas basados en cuentas, hemos utilizado la matriz de co-ocurrencia de ngramas para el entrenamiento de los embeddings neuronales.

Así, hemos propuesto evaluar un sistema de reconocimiento de idioma que esté basado en la representación vectorial de ngramas fonéticos, que por una parte modele las representaciones basándose en la información de contexto local (Mikolov y col., 2013c) y por otra las modele en base de su co-ocurrencia dentro del corpus (Pennington, Socher y Manning, 2014). En el caso de estudios a nivel fonético, como el nuestro, donde la sintaxis y la semántica tienden a ser poco influyentes en la tarea de reconocimiento, entendemos que el uso de ngramas-fonéticos como parámetros en la tarea LID pueden caracterizar de manera más eficiente a los embeddings entrenados si son tratados de forma similar a los i-Vectores, como vectores continuos que contengan la información más representativa de un idioma en baja dimensión. Lo que buscamos es que los embeddings de un idioma estén proyectados de una manera particular y que la proyección resulte discriminativa en comparación con las proyecciones del resto de idiomas.

En general, los modelos con los que se obtienen los ENs están caracterizados por la relación existente entre el elemento de entrada, su contexto y la representación del

FIGURA 7.1: Ejemplo de un vector de embedding de un ngrama-fonético

FIGURA 7.2: Obtención de los embeddings neuronales

contexto (Lai y col., 2015), y por ello se utiliza la probabilidad condicional $p(c|\omega)$, que predice el elemento siguiente en aparecer ω basado en su contexto c .

De acuerdo a estos conceptos, para la creación de los ENs seguimos el siguiente procedimiento:

- A partir del conjunto de ngramas-fonéticos utilizados para el entrenamiento de los ENs y aplicando la codificación 1-de-N en la capa de entrada de la red neuronal (Salamea, D'Haro y Cordoba, 2018), se le asocia a cada uno de los ngramas-fonéticos un vector característico de dimensión m (Figura 7.1) que contiene los valores generados en la capa de estado de la red (Figura 7.2) que es la representación vectorial del ngrama-fonético de entrada. El conjunto de vectores de ngramas-fonéticos genera una matriz característica que agrupa a las representaciones vectoriales de todos los ngramas-fonéticos utilizados para el entrenamiento de los ENs. Posee $[V * m]$ parámetros que deben ser entrenados, siendo V el tamaño del vocabulario y m el tamaño de la capa de estado.

- En función del tamaño de ventana v , se usan las representaciones vectoriales (la propia del n-grama analizado y las de sus vecinos) para identificar la mayor probabilidad de ocurrencia del próximo n-grama en aparecer. En el ejemplo mostrado en la Figura 7.2 se utiliza el modelo Skip-Gram con el que se busca maximizar la probabilidad de

ocurrencia del contexto de acuerdo al n-grama analizado. Para el caso de Skip-Gram, el próximo contexto en aparecer está definido por la representación vectorial del n-grama del contexto más probable dentro de la ventana v .

- Refiriéndonos nuevamente a la Figura 7.2, el ngrama analizado es n_i_j , el contexto está formado por los n-gramas j_o_n , u_n_i , n_a_d y i_j_o , y el embedding neuronal creado a partir del modelado Skip-Gram es la representación vectorial de i_j_o con la que se obtuvo la mayor probabilidad de ocurrencia.

- Para determinar el valor de probabilidad de ocurrencia entre las representaciones vectoriales del ngrama analizado y de los ngramas del contexto se utiliza una normalización Soft-max jerarquizada (Morin y Bengio, 2005) en la capa de salida de la red neuronal. Esto significa que el ngrama analizado se junta en pares con cada uno de los ngramas de contexto que están dentro de la ventana de análisis y mediante la función Soft-max jerarquizada se determina cual de los pares es el que presenta mayor probabilidad de ocurrencia.

- Este valor a la salida de la red junto con el valor esperado, son los que se utilizan para el proceso de aprendizaje de la red y en definitiva del entrenamiento de los embedding neuronales. El algoritmo de aprendizaje en la mayoría de los casos es el de retropropagación y minimización del error (Guo, 2013).

7.2. Embeddings Neuronales usados como vectores de parámetros

Inspirados en (Huang y col., 2012), que trabaja en tareas de procesamiento de lenguaje natural, donde la secuencia de palabras s se representa como una lista ordenada de vectores $x = (x_1, x_2, \dots, x_m)$ donde x_i es el embedding de la palabra i dentro de la secuencia, pensamos que el uso de secuencias de ngramas-fonéticos representadas por secuencias de embeddings pueden favorecer las tareas de reconocimiento de idioma.

7.2.1. Descripción del sistema

El sistema lo forman dos componentes, el Front-End que tiene una estructura igual a la del PPRLM (Zissman, 1996) y el Back-End que tiene una estructura similar a la de los sistemas basados en i-Vectores. En lo que se refiere al Front-End, partimos de las secuencias de fonemas obtenidas de los reconocedores fonéticos (Schwarz, 2009).

A estas secuencias de fonemas, las convertimos en secuencias de ngramas-fonéticos y a éstas últimas las transformamos en secuencias de embeddings neuronales (ENs). Las secuencias de ENs forman matrices bi-dimensionales (Figura 7.3), donde las filas,

FIGURA 7.3: Secuencias de ngramas-fonéticos usadas como vectores de parámetros en LID

de tamaño N (donde N corresponde al número de ngramas-fonéticos que forman la secuencia) representan la secuencia de ngramas-fonéticos de la frase y las columnas, de tamaño $V_e \times N_L$ (donde V_e es el tamaño del vector embedding y N_L es el número de idiomas a reconocer) definen el tamaño del vector *Total único agrupado* que agrupa a los vectores embedding de cada uno de los idiomas a reconocer para cada uno de los ngramas-fonéticos de la secuencia.

Para el caso de los ngramas-fonéticos no vistos en el entrenamiento de los ENs, les impusimos un vector embedding con valores cercanos a cero para que no afecten significativamente al valor generado por el *Embedding único agrupado*. Este procedimiento es útil para el caso de 2gramas, pero no para el caso de 3gramas-fonéticos, donde el porcentaje de unidades no vistas en el entrenamiento resulta ser mayor y significativo, generando secuencias de embeddings notoriamente mal representadas. Por esta razón, se ha considerado el estudio con unidades de 2gramas.

Estas secuencias se han utilizado como vectores de parámetros en el segundo componente del sistema, el Back-End, en donde, tras aplicarles una reducción de dimensionalidad mediante PCA, se utilizan para el entrenamiento de la matriz T y del UBM, a partir de los cuales se extraen los i-Vectores, que a su vez se usan para entrenar un clasificador MLR que incluye la calibración y la fusión de los scores. La estructura del sistema utilizado se muestra en la Figura 7.4.

Como se puede ver en la Figura 7.4, para obtener las secuencias de ENs, lo primero que se realiza es el entrenamiento de los ENs, que consiste en agrupar las secuencias de ngramas-fonéticos por idioma y entrenarlos por medio de la aplicación de Skip-Gram. Lo que se obtiene son los ngramas-fonéticos y sus respectivos vectores embedding. Se toman de uno en uno los ngramas-fonéticos de una secuencia, y se reemplaza por la

FIGURA 7.4: Sistema LID por medio de embeddings neuronales

concatenación de los vectores de embedding correspondientes a dicho ngrama-fonético en cada uno de los idiomas a reconocer (Embedding único agrupado) (Figura 7.3).

Otra opción que hemos considerado es que se entrenen las secuencias de ENs para cada uno de los idiomas de manera independiente *Embedding individual*, para luego proceder a su fusión. De esta manera la dimensión de los vectores es más reducida y más homogénea, al no unir información de idiomas diferentes.

Embeddings con información de contexto

Con el fin de que las secuencias de ENs incorporen más información relativa al idioma, hemos propuesto la inclusión de información de contexto por medio de los ENs de los ngramas-fonéticos vecinos en la secuencia. Para ello, tomamos en cuenta la hipótesis planteada por Mikolov (Mikolov y col., 2013a) que plantea que la representación vectorial obtenida a partir de modelos neuronales exhibe ciertas regularidades lingüísticas como la composición aditiva. La idea consiste en crear un embedding a partir de la suma de un porcentaje (el mayor) del embedding del ngrama-fonético analizado y de los porcentajes (menores) de los embeddings de los ngramas-fonéticos vecinos.

Así, para una primera evaluación, utilizamos la información del EN anterior y del posterior al EN analizado y los usamos tal como describimos en el siguiente ejemplo. Si tenemos la secuencia $u_n: n_i i_J$ el nuevo EN del 2grama-fonético n_i que reemplaza al EN del mismo ngrama-fonético, estaría formado por el 50 % de información del vector embedding de n_i y el 25 % de los vectores embedding de u_n y de i_J , tal como se muestra en la Figura 7.5.

En una segunda evaluación, ampliamos la información de contexto utilizando los dos ENs anteriores y posteriores al EN analizado. Así, la distribución de porcentajes que planteamos es del 50 % para el EN analizado, el 15 % de los ENs anterior y posterior al

FIGURA 7.5: Formación de embeddings de contexto

analizado y el 10 % de los ENs de los vecinos más lejanos, siguiendo la filosofía de la Figura 7.5.

7.2.2. Configuración del sistema

Al sistema lo evaluamos con dos tipos de configuraciones, la primera que llamamos *Embedding único agrupado* donde los vectores de embedding usados para la obtención de los i-Vectores están formados por la concatenación de los embeddings de cada ngrama-fonético para cada uno de los idiomas a reconocer, y la segunda que llamamos *Embedding individual* donde, las secuencias de ENs incluyen información únicamente de los vectores-embedding de un solo idioma, con lo cual generamos 6 sistemas (uno por cada idioma a reconocer) procediendo luego a su fusión.

7.2.3. Resultados

Embedding único agrupado

La configuración de inicio la llevamos a cabo modelando las representaciones vectoriales de los ngramas-fonéticos por medio de *Skip-Gram*, con un tamaño de ventana de 8, un muestreo negativo NS=25 y con 10 iteraciones. El tamaño del vector de embedding es igual a 40, con lo cual, al tener 6 idiomas a reconocer, la dimensión del vector para el

Embedding único agrupado es de 240, utilizando 128 gaussianas y un PCA de dimensión 100. Con esta configuración hemos obtenido un C_{avg} de 26,65 %.

Respecto al PCA, hemos evaluado el sistema con PCA=120, 100 y sin PCA. Los mejores resultados se han obtenido sin utilizar PCA, al presentar una mejora del 3,96 % en relación a la configuración inicial y, por ello, decidimos que la experimentación continuara sin realizar reducciones de dimensionalidad.

Respecto al número de gaussianas utilizadas llevamos a cabo un barrido desde 16 hasta 512 gaussianas, obteniendo el óptimo para 512 gaussianas con un C_{avg} de 24,69 %, representando una mejora del 7,35 % en relación a la configuración de inicio.

Embedding individual

En esta configuración, en vez de utilizar un vector en el que se concatenan los vectores de embedding de cada idioma se utilizan vectores de embedding individuales para cada ngrama-fonético de un idioma determinado.

Para esta configuración, igual que sucedía para la configuración *Embedding único agrupado* y dado que los vectores son de menor dimensión, no ha resultado necesario aplicar PCA.

Esta configuración la usamos para evaluar la idea de los embeddings de contexto. Hemos evaluado, tanto, la configuración *B-A-B* (25-50-25) que combina la información embedding del ngrama-fonético analizado *B* y de sus dos vecinos más cercanos *A*, como para la combinación *C-B-A-B-C* (10-15-50-15-10) que incluye a dos vecinos más alejados.

En experimentos previos realizados con 32 gaussianas (con el objetivo de disminuir el tiempo de procesamiento) se determinó que la configuración *B-A-B* resultó la mejor y, por ello, nos quedamos con ella para los experimentos posteriores. Respecto al tamaño de los vectores de embedding, hemos evaluado los tamaños 10, 20 y 40, encontrando el mejor resultado con 10. De esta manera, la configuración óptima del sistema *Embedding individual* se definió con los parámetros que se muestran en la Tabla 7.1 obteniendo un C_{avg} de 24,38 % utilizando únicamente uno de los modelos, en concreto el generado para el idioma Vasco. Como podemos ver, y como era previsible, esta técnica es mucho más eficiente, al conseguir mejoras ya con uno solo de los modelos.

Aplicando la misma técnica y configuración al resto de los idiomas, obtenemos los siguientes resultados.

Cabe destacar la mejora que se consigue con la fusión de todos los idiomas utilizando la técnica *Embedding individual*. En la Tabla 7.3 se comparan los resultados con los obtenidos con la técnica *Embedding único agrupado*. La mejora es del 24 % relativo, lo que demuestra la bondad del enfoque.

CUADRO 7.1: Parámetros óptimos Embedding individual.

Parámetro	Óptimo
Ventana	10
Muestreo negativo (NS)	25
Iteraciones	10
Vector-embeddings	10
Gaussianas	512

CUADRO 7.2: C_{avg} para cada idioma y Fusión para embeddings individuales con Skip-Gram y contexto B-A-B (25-50-25).

Sistema	C_{avg}
Vasco	24,38
Catalán	24,40
Inglés	24,51
Gallego	25,14
Portugués	26,23
Español	25,44
Fusión	18,70

CUADRO 7.3: Comparación de las técnicas *Embedding único agrupado* y *Embedding individual*.

	C_{avg}	Mejora %
Embedding único agrupado	24,69	
Embedding individual	18,70	24,3

Utilización de GloVe para generar los ENs

Dado que los mejores resultados se han obtenido utilizando la técnica *Embedding individual* usando la configuración B-A-B, hemos seguido utilizando esta configuración para una nueva alternativa que hemos considerado para crear los ENs, la técnica *GloVe*, y así comparar el rendimiento de éstos con los obtenidos utilizando *Skip-Gram*.

GloVe son modelos globales de representación vectorial que toman como base a la matriz de co-ocurrencia de ngramas-fonéticos a lo largo del corpus, aplicando un modelo de mínimos cuadrados en el entrenamiento.

El objetivo del entrenamiento de *GloVe* es aprender representaciones vectoriales de tal manera que el producto escalar sea igual al logaritmo de la probabilidad de co-ocurrencia de los ngramas-fonéticos. La matriz de co-ocurrencia contiene cuán frecuentemente co-ocurren los ngramas entre ellos en un corpus. Lo que busca el modelo *GloVe* es normalizar las cuentas y luego suavizarlas, esperando conseguir una representación vectorial sin grandes saltos de valores.

La diferencia entre el modelo *GloVe* y el *Skip-Gram* está en el *contexto de ngrama* considerado para la generación del modelo (entendemos como *contexto de ngrama* a los ngramas que están alrededor del ngrama analizado dentro de una ventana de tamaño v). En el caso de *GloVe* el *contexto de ngrama* se obtiene considerando todo el corpus, ya que la matriz de co-ocurrencia, que es la base a partir de la cual se generan las representaciones vectoriales, se forma a partir del número de cuentas de co-ocurrencia de los ngramas analizados y sus respectivos *contextos de ngrama* dentro del corpus. Por otro lado, en el caso de *Skip-Gram* el *contexto de ngrama* es local, lo que implica que, dentro de una frase, al ngrama analizado le corresponde el *contexto de ngrama* formado únicamente por sus vecinos más cercanos que pueden ser de mayor o menor número en función del tamaño de la ventana v .

Para evaluar el comportamiento de los modelos *GloVe* (Pennington, Socher y Manning, 2014), se ha experimentado con diferentes valores, tanto para definir el valor óptimo del tamaño de vector, el tamaño de la ventana de contexto y el número de iteraciones requeridas. Después de evaluar el sistema usando varios tamaños de vector desde 20 hasta 110, hemos encontrado el mejor resultado con un tamaño de vector igual a 40. En lo que se refiere al tamaño de la ventana de contexto hemos evaluado el sistema con valores entre 2 y 10 encontrando el óptimo para una ventana igual a 4. El número de iteraciones lo hemos establecido a 30.

En relación a la reducción de dimensionalidad y al número de gaussianas usado para la generación de los i-Vectores hemos encontrado los mejores resultados cuando no hemos aplicado PCA a la vez que hemos usado 512 gaussianas (al igual que para *Skip-Gram*).

CUADRO 7.4: C_{avg} para cada idioma y su Fusión para embeddings individuales con GloVe y contexto B-A-B (25-50-25).

Sistema	C_{avg}	Mejora %
Vasco	23,20	4,8
Catalán	21,59	11,5
Inglés	23,17	5,5
Gallego	22,75	9,5
Portugués	27,31	-4,1
Español	21,96	13,7
Fusión	16,70	10,7

CUADRO 7.5: Sistema Acústico de referencia con Coeficientes MFCCs para KALAKA-3.

	C_{avg}	Mejora %
Sistema Acústico	7,60	
Sistema Embedding individual (Skip-Gram)+Acústico	5,40	28,9
Sistema Embedding individual (GloVe)+Acústico	5,01	34,1

En la Tabla 7.4 se muestran los resultados obtenidos en C_{avg} al generar los embeddings con modelos GloVe usando la configuración B-A-B. Los porcentajes de mejora se encuentran referidos a los resultados mostrados en la Tabla 7.2 donde los embeddings se generaron usando Skip-Gram.

Hemos realizado pruebas para la configuración C-B-A-B-C (10-15-50-15-10) que incorpora la información de los dos vecinos anteriores y posteriores al n-grama analizado pero en ningún caso mejoró a los resultados obtenidos con la configuración B-A-B (25-50-25).

Con los resultados de la Tabla 7.4 se puede ver que el uso de modelos GloVe representan una mejora del 10,7 % de C_{avg} relativo en comparación con los resultados obtenidos con Skip-Gram.

7.2.4. Fusión con el sistema acústico basado en MFCCs

Por último, hemos llevado a cabo la fusión del sistema *Embedding individual* tanto de los embeddings generados con Skip-Gram como de los generados con GloVe con el sistema acústico basado en MFCCs que se ha venido utilizando a lo largo de la tesis. El resultado se presenta en la Tabla 7.5.

7.3. Aplicación de Embeddings neuronales para optimizar las secuencias de ngramas-fonéticos

En esta sección, dentro del estudio de los embeddings neuronales (ENs) nos propusimos evaluar en qué medida la información numérica (representación vectorial) referida a códigos (ngramas-fonéticos) representada por los ENs podría mejorar las secuencias fonéticas con las que se entrena la red neuronal para la obtención de los modelos de lenguaje, retomando la línea de experimentación del capítulo 6.

Como veíamos al estudiar el sistema de reconocimiento basado en RNNLM-P del capítulo 6, el uso de ngramas-fonéticos de orden superior provoca dispersión de la información debido al incremento del tamaño del vocabulario. Hemos decidido aprovechar la potencia de los ENs para reemplazar los ngramas-fonéticos más dispersos y menos significativos por otros que sean similares y más representativos al aparecer con más frecuencia en la base de datos, como veremos más adelante.

7.3.1. Descripción del sistema

La dispersión y la poca frecuencia de aparición de los ngramas-fonéticos de orden superior la abordamos reemplazando dichas unidades, representadas por sus ENs, por otras que estuvieran más cercanas en el espacio de representación vectorial (Mikolov y col., 2013c), quedando mejor representadas y estimadas en el idioma correspondiente al aparecer con más frecuencia.

Con este objetivo, se entrenan ENs que luego se utilizan para modificar y optimizar las secuencias de *ngramas-fonéticos*. Los ENs y los RNNLMs han sido entrenados con la misma información mediante un método supervisado, con la intención de que en sus resultados se encuentren implícitamente identificadas las características del idioma entrenado. Durante el entrenamiento de los ENs, cada ngrama-fonético de la secuencia se representa vectorialmente, se le aplica la técnica *Skip-Gram* y al contexto más probable se le asigna el vector de embedding característico para ese ngrama-fonético. Este mismo procedimiento se lleva a cabo para todos los ngramas-fonéticos de la secuencia y así, mientras más veces se repita uno u otro ngrama-fonético en la secuencia, el vector de embedding quedará cada vez mejor entrenado.

Para cada idioma, agrupamos en *clases* a los ngramas-fonéticos más cercanos entre sí, buscando que fueran los más similares en el espacio de los modelos de representación vectorial (VSMs). Además, considerando el número de apariciones de cada ngrama-fonético en el conjunto de datos de entrenamiento, identificamos cuáles son los ngramas-fonéticos más y menos representativos de un idioma. A continuación presentamos las

dos técnicas con las que hemos buscado optimizar las secuencias fonéticas.

Clustering con Embeddings Neuronales

Teniendo en cuenta que los ngramas-fonéticos menos característicos de un idioma no forman parte de grupos similares sino que tienden a estar dispersos, hemos creído que reemplazándolos por otros más cercanos que fueran más representativos de un idioma, mejoraríamos las características del modelo y por ende el proceso de clasificación y reconocimiento (Salamea, D'Haro y Cordoba, 2018).

Para definir si un ngrama-fonético es disperso o no, usamos la distancia escalar obtenida entre cada uno de los ENs de los ngramas-fonéticos aparecidos en la fase de entrenamiento. Así, utilizamos el algoritmo k-means, que compara el embedding del ngrama-fonético en cuestión con el embedding de cada uno de los otros, agrupando los ngramas-fonéticos con menos distancia entre sí.

Sustitución de unidades menos frecuentes usando embeddings y reglas heurísticas

Con el entrenamiento de los embeddings también se extrajo información de la frecuencia de aparición de cada uno de los ngramas-fonéticos. Así, a partir de los ENs entrenados con los textos originales utilizamos un valor umbral para determinar cuáles son los menos representativos de un idioma. Los que estén por debajo de dicho umbral se reemplazan por unidades *similares* pero más representativas, cuyo número de apariciones es mayor al umbral. Los criterios para decidir si una unidad es similar a otra se fundamentan en mantener el sonido alofónico de base y eliminar diferentes variaciones alofónicas. Para ello realizamos los siguientes cambios en los ngramas-fonéticos poco representativos:

- a. Eliminamos el elemento ":", que se refiere a una variante alofónica más larga o acentuada de la de base.
- b. Se han utilizado solo minúsculas, para ello se transforman las mayúsculas por minúsculas.
- c. Los elementos alofónicos dobles se modificaron manualmente de acuerdo a la (Tabla 7.6):

El sistema empleado se muestra en la Figura 7.6, donde se puede ver que antes del entrenamiento y evaluación de la RNN, se lleva a cabo el entrenamiento de los ENs y el reemplazo de ngramas-fonéticos dispersos y poco frecuentes. Los valores de entropía obtenidos por la RNN para cada fichero e idioma son las entradas de un clasificador logístico multi-clase. Este mismo clasificador es el que permite realizar la fusión de los tres modelos del reconocedor de fonemas. Con la Figura 7.6, además se puede observar

CUADRO 7.6: Elementos alofónicos dobles.

Original	Reemplazado por:
Au	a
Eu	e
Jj	j
Ou	o
Ts	t
Ss	s

FIGURA 7.6: Sistema de reconocimiento basado en RNNLMs con ENs

que el trabajo desarrollado se centró en el Back-End, justo a partir de las secuencias fonéticas generadas por el reconocedor del Front-End.

7.3.2. Configuración del sistema

En lo que se refiere al modelo utilizado para entrenar los ENs, utilizamos Skip-Gram (Guthrie y col., 2006) basándonos en los buenos resultados obtenidos al entrenar un sistema LID basado en i-Vectores donde usamos los ENs de 2gramas como vectores de parámetros. Para ello, cada 2grama se reemplaza por su respectivo EN y las secuencias de vectores de embedding se utilizan como vectores de parámetros para entrenar una matriz de variabilidad total T y un modelo universal UBM , que, a su vez se utilizan para obtener los i-Vectores respectivos. Estos a su vez, se utilizan para entrenar un clasificador multi-clase de regresión logística donde se calibran y fusionan los scores (Brummer y col., 2007). Con esta evaluación hemos comprobado que Skip-Gram supera a C-Bow con hasta un 15,64 % de mejora relativa. Por otro lado hemos podido verificar otros trabajos donde se han llevado a cabo comparaciones entre las dos técnicas como en el trabajo de Xu (Xu, 2016) donde el rendimiento de Skip-Gram supera al de C-Bow en

el modelado de representaciones vectoriales. Por estas razones, se mantuvo el método Skip-Gram para el resto de experimentos y en el sistema final.

El siguiente paso consiste en definir los parámetros óptimos de los ENs tanto para 2gramas como para 3gramas, definiendo en primer lugar el tamaño del vector embedding, luego el tamaño de la ventana, el número de iteraciones y por último el NS.

En el caso de 2gramas, el tamaño óptimo del vector embedding ha sido establecido en 40, luego de evaluar tamaños de 10, 20, 40, 50 y 80. El tamaño óptimo de ventana es igual a 8, después de evaluar el sistema con ventanas de 5, 8 y 10. El número de iteraciones es igual a 10 tras comprobar que para un número mayor de iteraciones las mejoras ya no resultan significativas. La optimización del modelo Skip-Gram usado en 2gramas se ha llevado a cabo con el NS, evaluando el sistema para NS=15, 25 y 50, encontrando el mejor resultado para NS=25. En resumen, la configuración óptima de los ENs para 2gramas se estableció con: un vector embedding=40, una ventana=8, un NS=25 y con 10 iteraciones.

En el caso de 3gramas, el tamaño óptimo del vector se ha establecido en 110, tras evaluar los tamaños 40, 100, 110, 120, 150, 200 y 300. El tamaño óptimo de ventana es igual a 8, tras evaluar el sistema con ventanas de 4, 8 y 10. El número de iteraciones ha sido establecido a 10 tras evaluar el sistema con 1, 5, 10, 15 y 20 iteraciones. En lo que se refiere al NS evaluamos el sistema para NS=25, 50, 75, 100, 150, 200 y 300, encontrando un óptimo en NS=150. En resumen, la configuración óptima de ENs para 3gramas se estableció con: vector embedding=110, el tamaño de ventana=8, un NS=150 y utilizando 10 iteraciones.

7.3.3. Resultados

Aplicación del Clustering con Embeddings Neuronales

Como se ha descrito en la sección anterior, la primera técnica consideraba se basa en reemplazar los ngramas-fonéticos más alejados de sus centroides por otros similares más cercanos. El algoritmo de clustering utilizado asume que los centroides están rodeados de vecinos con densidades locales más bajas, y que ellos están a una distancia relativamente más grande de vecinos con una densidad más alta (Wang y col., 2016).

La densidad baja de ngramas-fonéticos ρ_i se define tal como se describe en la ecuación 7.1.

$$\rho_i = \sum_j x(d_{ij} - d_c) \quad (7.1)$$

CUADRO 7.7: Aplicación del clustering con embeddings

ngramas-fonéticos	C_{avg}	Mejora %
2gramas-RNNLM-P (base)	12,09	
2gramas-reducción 20 %	11,49	5,0
3gramas-RNNLM-P (base)	12,02	
3gramas-reducción 10 %	11,92	1,0
Fusión 2-3gramas-RNNLM-P (base)	11,15	
Fusión reducción	10,87	2,5

Donde d_{ij} es la distancia entre dos puntos y d_c es una distancia base y $x = 1$, si $d_{ij} < d_c$ o $x = 0$, en caso contrario.

Por otra parte, los puntos de alta densidad δ_i se definen como: mínimo de d_{ij} , si $\rho_i < \rho_{max}$ ó como máximo de d_{ij} , en caso contrario.

La distancia $d_{ij} - d_c$ con la que se decide si un ngrama es disperso o no se definió por medio de una constante que determina el porcentaje de vocabulario que se reduce. Así, el porcentaje de vocabulario reducido fue directamente proporcional a la distancia de decisión. Aplicando este criterio, pudimos encontrar que para 2gramas y 3gramas-fonéticos, los óptimos se encontraban reduciendo el vocabulario en un 20 % y un 10 %, respectivamente. En la Tabla 7.7 se muestran los resultados.

Aplicación de la sustitución de unidades menos frecuentes usando embeddings y reglas heurísticas

En esta técnica utilizamos la frecuencia de ocurrencia de los ngramas-fonéticos para definir los que se deben reemplazar. Para ello, utilizamos un valor de umbral que selecciona los ngramas-fonéticos menos frecuentes y los reemplaza por otros similares más frecuentes. Así, por ejemplo, un umbral de 3, implica que todos los ngramas-fonéticos que han aparecido hasta dos veces se consideran como no-frecuentes y, por lo tanto, son reemplazados.

De la experimentación realizada, encontramos que para 2gramas y 3gramas-fonéticos, los óptimos se obtuvieron con umbrales de 3 y 2, respectivamente. En la Tabla 7.8 se muestra lo descrito. Podemos comprobar que esta técnica aporta mejores resultados que la basada en clustering.

7.3.4. Fusión con el sistema acústico basado en MFCCs

Por último, usamos nuestro sistema para mejorar el sistema acústico basado en MFCCs descrito en el Capítulo 3. En la Tabla 7.9 se muestran los resultados.

CUADRO 7.8: Aplicación de la sustitución de unidades frecuentes

ngramas-fonéticos	C_{avg}	Mejora %
2gramas-RNNLM-P (base)	12,09	
2gramas-umbral 3	11,14	7,9
3gramas-RNNLM-P (base)	12,02	
3gramas-umbral 2	11,36	5,6
Fusión 2-3gramas-RNNLM-P (base)	11,15	
Fusión reducción	10,09	9,5

CUADRO 7.9: Fusión del sistema Embedding *Reemplazo menos frecuentes* con Acústico.

ngramas-fonéticos	C_{avg}	Mejora %
Sistema Acústico	7,60	
Sistema Embedding <i>Reemplazo menos frecuentes</i> con Acústico	5,04	33,7

De las Tablas 7.8 y 7.9 se puede observar que las técnicas en las que hemos usado embeddings para la tarea LID mejoran los resultados obtenidos al utilizar RNNs en la creación de modelos de lenguaje usando ngramas-fonéticos al optimizar las secuencias fonéticas con las que se entrena la red.

Por otro lado, cuando hemos usamos los embeddings neuronales como parámetros en la tarea LID, hemos podido experimentar con varios métodos y aunque hemos obtenido mejoras interesantes con cada uno de ellos, la mejora efectiva la hemos conseguido al fusionar los resultados de esta técnica con los del sistema acústico de referencia.

7.3.5. Conclusiones y aportaciones

Hemos propuesto el uso de los modelos de representación vectorial a nivel de ngramas-fonéticos para tareas LID basados en las mejoras obtenidas con la generación de RNNLMs a nivel fonotáctico. Como vectores de parámetros: Hemos propuesto dos técnicas:

- Embedding único agrupado - 24,69 % de C_{avg} ,
- Embedding individual - 18,7 % de C_{avg} (24,3 % mejor).

Si bien los resultados no son los mejores, sí que generan información complementaria importante en la fusión con el RNNLM-P y el acústico de referencia, consiguiendo hasta un 5,01 % de C_{avg} .

Para optimizar las secuencias usadas para entrenar los RNNLM-P hemos propuesto dos técnicas:

- Clustering automático de las representaciones vectoriales de los ngramas fonéticos existentes en las secuencias

- Reemplazo de ngramas-fonéticos menos frecuentes con otros de mayor frecuencia de ocurrencia.

Las aportaciones fueron las siguientes:

- Adaptación de los modelos de representación vectorial (diseñados a nivel léxico) al ámbito fonético-fonotáctico.

- Uso de embeddings neuronales fonéticos como vectores de parámetros en tareas LID

- Uso de embeddings neuronales fonéticos como optimizadores de las secuencias de entrenamiento de RNNLMs.

Capítulo 8

CONCLUSIONES, LINEAS FUTURAS Y PUBLICACIONES

8.1. Conclusiones

Como hemos podido comprobar, las técnicas a nivel fonético y fonotáctico propuestas en esta tesis para tareas de reconocimiento de idioma han aportado mejoras importantes.

Hablando ya propiamente del reconocimiento de idioma, que es el ámbito de investigación que ha motivado la presente tesis, y en el que hemos considerado a la información de contexto de los fonemas en un idioma como elemento central de estudio, hemos presentado varias propuestas, tanto a nivel de parámetros (posteriorgramas, los PLLR de estado y su variante los SDPC y los ngramas-fonéticos) como a nivel de técnicas (RNNLMs-P, embedding neuronales, Embedding único agrupado y Embeddings individuales), todas con el objetivo de mejorar la tasa de reconocimiento. Así, de acuerdo a los resultados obtenidos a lo largo de la experimentación, vamos a presentar un resumen de las conclusiones que consideramos más relevantes para cada una de las técnicas propuestas.

Respecto a la primera línea de investigación basada en utilizar coeficientes *posteriorgramas* e información discriminativa de idioma, nuestra aportación se encuentra en la incorporación de información discriminativa de idiomas en dichos posteriorgramas, utilizando un sistema de reconocimiento basado en i-Vectores para este cometido. La información de contexto de una frase aprovechada por los posteriorgramas junto con la incorporación de información discriminativa de idioma supuso una mejora significativa en el sistema LID. Con la aplicación de esta idea se llegó a obtener hasta un 11,1 % de mejora relativa con el factor de suavizamiento propuesto respecto al sistema de referencia basado en posteriorgramas. Por otro lado, al fusionarlo con el sistema acústico de referencia se ha obtenido un C_{avg} de hasta un 5,21 % que representa una mejora relativa del

8,4 %. Por lo tanto, podemos concluir de que la información de contexto sobre vectores de parámetros contribuye a la mejora del rendimiento de los sistemas de reconocimiento.

En la segunda línea de investigación basada en la utilización de parámetros PLLRs a nivel de estado y la obtención de una configuración estándar para su uso, hemos utilizado el mismo sistema de i-Vectores y en lugar de los posteriorgramas, hemos planteado la utilización de los parámetros PLLR. Hemos propuesto su utilización a nivel de estado en vez de a nivel de fonema, lo que ha mejorado la tasa de reconocimiento, aunque es cierto que a costa de tener que utilizar técnicas de reducción de dimensionalidad debido al incremento de la dimensión del vector de parámetros (el triple dado que usamos los tres estados del reconocedor). Por otra parte, hemos utilizado los parámetros PLLR para formar lo que hemos llamado parámetros SDPC, que además incluyen información del contexto temporal gracias a la aplicación y optimización de la técnica SDC en los PLLR. Como hemos podido comprobar, la aplicación de los SDPC mejoran los resultados en el sistema LID, aunque también observamos que el sistema es sensible a los cambios de parámetros del SDC, en especial los parámetros M (dimensión del vector tras la PCA) y K (número saltos de contexto que se consideran), por lo que los óptimos de uno y otro van a depender de la tarea a desarrollar. En cualquier caso, hemos obtenido óptimos estables para los parámetros D y P , y, en general, se obtienen resultados óptimos y similares con valores del mismo entorno del producto $M(K + 1)$. El mejor resultado para el sistema basado en fonemas fue del 6,62 % y para el basado en estados del 6,35 %. A continuación, al hacer la fusión de todos los sistemas PLLR con el módulo acústico los resultados tienden a igualarse, aunque obtenemos un resultado final del 3,19 %, que supone una mejora drástica del 58 % frente al sistema de referencia acústico (tasa del 7,60 %).

A continuación, y en relación a la tercera línea de investigación propuesta, hemos estudiado un sistema de reconocimiento de idioma que utiliza modelos de lenguaje basados en redes neuronales recurrentes pero en el ámbito fonético y fonotáctico. Nos interesaron particularmente las RNNs debido a que se utiliza información de contexto para la generación de los modelos de lenguaje. Por esta razón, decidimos utilizar parámetros que tuvieran en cuenta la información de contexto en la secuencias de fonemas, en concreto los *ngramas-fonéticos*, con lo que las secuencias de fonemas se convirtieron en secuencias de unidades más extensas que no llegan a ser de nivel léxico. El uso de estas nuevas unidades en la creación de los modelos de lenguaje, permitió, por una parte, incrementar el inventario de unidades fonéticas con lo que aumentamos el nivel de información del idioma mejorando de este modo la tarea LID y, por otra parte, introducimos unidades que de por sí poseían información de contexto, que era la información que queríamos aprovechar.

La tarea no resultó sencilla por dos motivos. En primer lugar, el aumento en el número de unidades produjo una mayor dispersión de la información y muchas de las unidades terminan con una estimación pobre. En segundo lugar, resultó necesario configurar los parámetros de la red para cada caso y orden de ngrama. Se tuvieron que modificar las mejores condiciones de la red, partiendo del número óptimo de neuronas en la capa de estado en función de los vocabularios creados, pasando por el número de clases en la capa de salida hasta la parametrización de los factores de regularización de la propia red. Con la red reconfigurada y adaptada a las nuevas unidades de entrada, los *ngramas-fonéticos*, se logró mejorar el rendimiento de la red en relación a utilizar fonemas como unidades de entrada. Además, introducimos el concepto de probabilidad mínima para disminuir la penalización aplicada por la RNN a los *ngramas-fonéticos* no vistos en el entrenamiento y con ello logramos mejorar hasta en un 13 % el rendimiento de la red comparándola nuevamente con el sistema basado en fonemas. Nos centramos en *ngramas-fonéticos* de dos y tres elementos, dado que para unidades de mayor orden la dispersión crecía tanto que las estrategias utilizadas no presentaron mejores resultados. Los resultados presentados son mejores que los obtenidos con el sistema de referencia basado en PPRLM en un 3,6 % y, en cualquier caso, hemos comprobado que ambos sistemas aportan en la fusión con el sistema acústico y que la fusión de los tres sistemas es la que aporta el mejor resultado con una mejora del 36,8 % en relación al sistema acústico.

Respecto a la cuarta y última línea de investigación que se presenta en esta tesis, basada en la utilización de *ngramas fonéticos* para la generación de RNNLMs, hemos propuesto la idea de utilizar la representación vectorial de los *ngramas-fonéticos* dentro de la red para utilizarlos en tareas LID en un ámbito fonético y fonotáctico (*ngramas fonéticos*) a diferencia del método normalmente utilizado a nivel léxico para tareas de procesamiento de lenguaje natural. Lo primero que nos planteamos fue usar los embeddings neuronales (modelado de las representaciones vectoriales) como parámetros en nuestro sistema de i-Vectores. Así propusimos dos técnicas, la de *Embedding único agrupado* donde para cada *ngrama-fonético* los vectores de embedding se forman concatenando los embeddings de cada uno de los idiomas a reconocer, y la denominada *Embedding individual* en la que se utilizan vectores de embedding individuales, se entrenan modelos para cada idioma y se fusionan los resultados obtenidos con cada uno de ellos. Esta última resultó claramente la mejor, con una mejora relativa del 24 %. Aunque los resultados obtenidos no llegaron a ser comparables a los obtenidos con las otras técnicas fonotácticas propuestas en la tesis, sí que presentan mejoras al realizar la fusión con esas otras técnicas (por ejemplo con la obtenida con los RNNLMs) y también aportan mejoras al fusionarlas con nuestro sistema acústico de referencia, consiguiendo hasta un 5,01 % en C_{avg} .

CUADRO 8.1: Resultados óptimos de las técnicas propuestas.

Sistema	C_{avg}
Acústico	7,60
Posteriorgrama (Ref)	9,85
Ranking	8,76
SDPCs-Estados	3,19
PPRLM	11,57
RNNLM-P	11,15
Embedding-GloVe	16,70

Por último, buscamos mejorar las secuencias de ngramas-fonéticos utilizadas para el entrenamiento de los RNNLMs modificando aquellos ngramas-fonéticos con pocas apariciones en el entrenamiento o aquellos cuya representación en el espacio vectorial resultare *lejana* en relación con el conjunto de datos. Así hemos propuesto dos técnicas, la primera mediante un clustering automático de las representaciones vectoriales de los ngramas-fonéticos existentes en las secuencias y la segunda, reemplazando los ngramas-fonéticos menos frecuentes con otros de mayor frecuencia de ocurrencia. De las dos técnicas, los resultados resultaron ser parecidos, aunque la sustitución de unidades frecuentes produjo los mejores resultados. La evaluación de esta técnica se realizó únicamente para ngramas-fonéticos de dos elementos dado que para los de tres elementos, el número de unidades crecía mucho generando matrices con muchos valores nulos.

En la Tabla 8.1 se presenta un resumen de los resultados óptimos para cada una de las técnicas propuestas en la Tesis:

En tanto que en la Tabla 8.2 se presenta un conjunto con las fusiones más destacadas.

De lo que se puede ver en la Tabla 8.2 se puede concluir que cada una de las técnicas propuestas en esta tesis genera información que es complementaria de las demás, dado que en la fusión se obtienen mejoras en la tareas de reconocimiento. De entre ellas, cabe resaltar los resultados obtenidos con *Embedding-GloVe* para los que a pesar de que como sistema individual obtienen los peores resultados, si que producen mejoras cuando se fusiona con los otros sistemas. Por otra parte, hemos decidido no incluir en el resumen de fusiones a la técnica *SDPCs-Estados+Acústico* dado que por sí misma genera una tasa de reconocimiento tan alta que no permite identificar las diferencias entre el resto de las técnicas presentadas en esta tesis. En una evaluación de todas las técnicas fusionadas salvo *SDPCs-Estados+Acústico* se obtiene hasta un 4,61 % de C_{avg} en la tarea de reconocimiento.

CUADRO 8.2: Fusión de las diferentes técnicas propuestas *Fusión de las diferentes técnicas propuestas*

Sistema	C_{avg}
Posteriorgrama+Acústico	5,69
Ranking+Acústico	5,21
SDPCs-Estados+Acústico	3,19
RNNLM-P+PPRLM	10,74
PPRLM+Acústico	5,10
RNNLM-P+Acústico	5,06
RNNLM-P+PPRLM+Acústico	4,80
Embedding-GloVe+Acústico	5,01
RNNLM-P+PPRLM+Ranking+Acústico	4,72
RNNLM-P+PPRLM+Ranking	7,92
RNNLM-P+PPRLM+Embedding-GloVe+Acústico	4,69
RNNLM-P+PPRLM+Embedding-GloVe	10,01
RNNLM-P+PPRLM+Ranking+Embedding-GloVe+Acústico	4,61

8.2. Líneas futuras

En relación al uso de información discriminativa dentro los RNNLMs proponemos la inclusión de umbrales de decisión aplicados a los valores discriminativos de las plantillas, de forma que únicamente los n-gramas con un número mínimo de repeticiones vean modificadas las cuentas de sus posteriorgramas. En esta misma línea, consideraremos la creación de nuevas plantillas en las que el valor discriminativo pueda ser calculado a partir de nuevas fórmulas pudiendo también utilizar umbrales. Finalmente, también trabajaremos en ampliar esta técnica utilizando un sistema tipo PPRLM en el que tengamos no sólo un reconocedor de fonemas si no que podamos usar varios reconocedores en paralelo (e.g. aprovechando también los modelos de Checo o Ruso que viene incluido con el reconocedor de la Universidad de Brno).

En relación a los SDPCs investigaremos nuevas técnicas para reducir la redundancia de información existente entre tramas adyacentes en el momento de usar parámetros basados en PLLRs. También se pretende reducir el número total de estados eliminando los fonemas menos frecuentes.

En relación a los RNNLMs basados en ngramas-fonéticos esperamos mejorar su rendimiento explorando nuevas formas de introducir información discriminativa de idioma en la red neuronal, de manera que podamos obtener modelos de lenguaje más robustos. Respecto a la representación vectorial de ngramas-fonéticos estudiaremos otras técnicas de clustering que permitan una reducción de vocabulario más eficiente.

Por otra parte, se pueden evaluar técnicas automáticas de configuración de parámetros para sistemas multivariantes como los de las *redes neuronales* para contrastar con los óptimos obtenidos en esta tesis.

Otra de las líneas futuras de investigación es la representación vectorial a nivel de trama. Para ello sería necesario aplicar técnicas de reducción de dimensionalidad dado el gran volumen de información que se vería involucrado.

8.3. Publicaciones

8.3.1. Publicaciones en Revistas Internacionales:

- Salamea Palacios, C., D'Haro Enríquez, L. F., Córdoba Herralde, R. (2018). Language Recognition Using Neural Phone Embeddings and RNNLMs. IEEE Latin America Transactions. (Aceptado y pendiente de publicación).

- San-Segundo, R., Echeverry-Correa, J. D., Salamea Palacios, C. R., and Pardo, J. M. (2016). Human Activity Monitoring Based on Hidden Markov Models Using a Smartphone. IEEE Instrumentation & Measurement Magazine. 19-6, pp. 27-21.

- San-Segundo, R., Echeverry-Correa, J. D., Salamea-Palacios, C., Lutfi, S. L., and Pardo, J. M. (2016). I-Vector analysis for gait-based person identification using smartphone inertial signals. Pervasive and Mobile Computing. pp. 140-153.

8.3.2. Publicaciones en Revistas Nacionales:

- Salamea Palacios, C. R., D'Haro Enríquez, L. F., Córdoba Herralde, R. D., and Caraballo Morcillo, M. Á. (2013). Incorporation of discriminative n-grams to improve a phonotactic language recognizer based on i-vectors. Procesamiento del lenguaje natural, (51), pp. 145-152.

8.3.3. Conferencias/Congresos:

- Salamea, C., D'Haro, L. F., de Córdoba, R., and San-Segundo, R. (2016). On the use of phone-gram units in recurrent neural networks for language identification. Odyssey 2016, pp. 117-123.

- Salamea, C., D'Haro, L. F., de Córdoba, R., and Montero, J. (2016). Phone-gram units in RNN-LM for language identification with vocabulary reduction based on neural embeddings. Iberspeech 2016, pp. 109-118.

- D'Haro, L. F., Cordoba, R., Salamea, C., and Echeverry, J. D. (2014, May). Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition. In Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2014. pp. 5342-5346.
- D'Haro Enriquez, L. F., Cordoba Herralde, R. D., Salamea Palacios, C. R., and Ferreiros López, J. (2014). Language recognition using phonotactic-based shifted delta coefficients and multiple phone recognizers. In Interspeech 2014. pp. 3042-3046.

Bibliografía

- Ambikairajah, E. y col. (2011). «Language identification: A tutorial». En: *Circuits and Systems Magazine, IEEE* 11.2, págs. 82 -108.
- Bell, Timothy C, John G Cleary y Ian H Witten (1990). *Text compression*. Prentice-Hall, Inc.
- Bengio, Y., P. Simard y P. Frasconi (1994). «Learning long-term dependencies with gradient descent is difficult». En: *Neural Networks, IEEE Transactions* 5.2, págs. 157-166.
- Bengio, Y. y col. (2006). «Neural probabilistic language models». En: *Innovations in Machine Learning*. Springer, págs. 137 -186.
- BenZeghiba, M. F., J.-L. Gauvain y L. Lamel (2009). «Language score calibration using adapted Gaussian back-end.» En: *Interspeech*, págs. 2191 -2194.
- Besacier, L. y col. (2014). «Automatic speech recognition for under-resourced languages: A survey». En: *Speech Communication* 56, págs. 85 -100.
- Bielefeld, B. (1994). «Language identification using shifted delta cepstrum». En: *Fourteenth Annual Speech Research Symposium*.
- Brummer, N. y D. A. van Leeuwen (2006). «On calibration of language recognition scores». En: *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The IEEE*, págs. 1 -8.
- Brümmer, N. y J. du Preez (2006). «Application-independent evaluation of speaker detection». En: *Computer Speech & Language* 20.2, págs. 230 -275.
- Brummer, N. y col. (2007). «Fusion of heterogeneous speaker recognition systems in the STBU submission for the NIST speaker recognition evaluation 2006». En: *Audio, Speech, and Language Processing, IEEE Transactions on* 15.7, págs. 2072 -2084.
- Campbell, W. M. y col. (2006a). «Support vector machines for speaker and language recognition». En: *Computer Speech & Language* 20.2, págs. 210 -229.
- Campbell, W. M. y col. (2006b). «Understanding scores in forensic speaker recognition». En: *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The IEEE*, págs. 1 -8.
- Caraballo, M. A. y col. (2010). «A discriminative text categorization technique for language identification built into a PPRLM System». En: *Proc. of VI Jornadas en Tecnología del Habla y II Iberian SLTech*, págs. 10 -12.
- Cavnar, W. B., J. M. Trenkle y col. (1994). «N-gram-based text categorization». En: *Ann Arbor MI* 48113.2, págs. 161 -175.

- Chang, F., LC. Chang y HL. Huang (2002). «Real-Time Recurrent Learning Neural Network for Stream-Flow Forecasting». En: *Hydrological Processes - Wiley Online Library*.
- Chen, P.-H., C.-J. Lin y B. Schölkopf (2005). «A tutorial on ν -support vector machines». En: *Applied Stochastic Models in Business and Industry* 21.2, págs. 111 -136.
- Chen, S. F. y J. Goodman (1996). «An empirical study of smoothing techniques for language modeling». En: *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, págs. 310 -318.
- Chen, Xie y col. (2015). «Improving the training and evaluation efficiency of recurrent neural network language models». En: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, págs. 5401-5405.
- Chitarroni, H. (2002). «La regresión Logística». En: *Instituto de Investigación en Ciencias*.
- Cordoba, R. y col. (2007). «Language Identification based on n-gram Frequency Ranking». En: *8Th Annual Conference of the International Speech Communication Association, Interspeech*. Vol. 3, págs. 1921 -1924.
- Cumani, S. (2012). «Speaker and Language Recognition Techniques». Tesis doct. Politecnico di Torino.
- Dahl, G. E. y col. (2012). «Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition». En: *Audio, Speech, and Language Processing, IEEE Transactions on* 20.1, págs. 30 -42.
- Davis, S. y P. Mermelstein (1980). «Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences». En: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 28.4, págs. 357 -366.
- Dehak, N. y col. (2009). «Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification.» En: *Interspeech*. Vol. 9, págs. 1559 -1562.
- Dehak, N. y col. (2011a). «Front-end factor analysis for speaker verification». En: *Audio, Speech, and Language Processing, IEEE Transactions on* 19.4, págs. 788 -798.
- Dehak, N. y col. (2011b). «Language Recognition via i-vectors and Dimensionality Reduction.» En: *Interspeech 2011*. Citeseer, págs. 857 -860.
- D'Haro, L. F. (2009). «Speed Up Strategies for the Creation of Multimodal and Multilingual Dialogue Systems». Tesis doct. PhD thesis, Universidad Politécnica de Madrid, Spain.
- D'Haro, L. F. y col. (2013). «Low-resource language recognition using a fusion of phone-me posteriorgram counts, acoustic and glottal-based i-vectors». En: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, págs. 6852 -6856.

- D'Haro, L. F. y col. (2014). «Language Recognition using Phonotactic-based Shifted Delta Coefficients and Multiple Phone Recognizers». En: *Fifteenth Annual Conference of the International Speech Communication Association*.
- Díez, M. y col. (2012). «On the use of phone log-likelihood ratios as features in spoken language recognition.» En: *Spoken Language Technology Workshop (SlT) IEEE*, págs. 274 -279.
- Diez, M. y col. (2013). «Using Phone Log-Likelihood Ratios as Features for Speaker Recognition». En: *ResearchGate 3*, págs. 1-5.
- Diez, Mireia y col. (2014). «On the complementarity of Short-Time Fourier Analysis Windows of Different Lengths for Improved Language Recognition». En: *Fifteenth Annual Conference or the International Speech Communitation Association*, págs. 3032 -3036.
- Dominguez, J. G. (2011). «Session variability compensation in automatic speaker and language recognition». Tesis doct. PhD thesis, Universidad Autónoma de Madrid.
- Duda, R. O., P. E. Hart y D. G. Stork (2012). *Pattern classification*. John Wiley & Sons.
- Goldberg, Yoav y Omer Levy (2014). «word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method». En: *arXiv preprint arXiv:1402.3722*.
- Gonzalez-Dominguez, J. y col. (2014). «Automatic language identification using Long Short-Term Memory recurrent neural networks». En: *Proc. Interspeech 2014*, págs. 2155 -2159.
- Goodman, J. (2001a). «Classes for fast maximum entropy training». En: *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. Vol. 1. IEEE, págs. 561 -564.
- Goodman, J. T. (2001b). «A bit of progress in language modeling». En: *Computer Speech & Language* 15.4, págs. 403 -434.
- Gordon, R. G. y B. F. Grimes (2005). *Ethnologue: Languages of the world*. Vol. 15. SIL international Dallas, TX.
- Guo, Jiang (2013). *BackPropagation Through Time*. HIT Social Computing e Information Retrieval Research Center.
- Guthrie, David y col. (2006). «A closer look at skip-gram modelling». En: *Proceedings of the 5th international Conference on Language Resources and Evaluation (LREC-2006)*, págs. 1-4.
- Hanson, S. J. y D. J. Burr (1990). «What connectionist models learn: Learning and representation in connectionist networks». En: *Behavioral and Brain Sciences* 13.03, págs. 471 -489.
- Hermansky, H. (1990). «Perceptual linear predictive (PLP) analysis of speech». En: *the Journal of the Acoustical Society of America* 87.4, págs. 1738 -1752.

- Hinton, Geoffrey y col. (2012). «Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups». En: *IEEE Signal Processing Magazine* 29.6, págs. 82-97.
- Hochreiter, S. (1998). «The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions». En: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*.
- Hochreiter, S. y col. (2001). «Gradient flow in recurrent nets: the difficulty of learning long-term dependencies». En: *A field guide to dynamical recurrent neural networks*. IEEE Press.
- Huang, Eric H. y col. (2012). «Improving Word Representations via Global Context and Multiple Word Prototypes». En: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*. ACL '12. Jeju Island, Korea: Association for Computational Linguistics, págs. 873-882.
- Hwang, Kyuyeon y Wonyong Sung (2016). «Character-Level Language Modeling with Hierarchical Recurrent Neural Networks». En: *arXiv preprint arXiv:1609.03777*.
- Jeffrey, Elman (1990). «Finding Structure in Time». En: *Cognitive Science - Wiley Online Library*.
- Jordan, Michael I (1986). «Serial order: A parallel distributed processing approach». En: *Tech. Rep. San Diego. University of California, Institute for Cognitive Science* 8604.
- Kajarekar, S. S. y A. Stolcke (2007). «NAP and WCCN: Comparison of approaches using MLLR-SVM speaker verification system». En: *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*. Vol. 4. IEEE, págs. 249-252.
- Kanagasundaram, A. y col. (2014). «I-vector based speaker recognition using advanced channel compensation techniques». En: *Computer Speech & Language* 28.1, págs. 121-140.
- Karsmakers, P. y col. (2007). «Multi-class kernel logistic regression: a fixed-size implementation». En: *International Joint Conference on ICNN*, págs. 1756-1761.
- Kenny, P., G. Boulianne y P. Dumouchel (2005). «Eigenvoice modeling with sparse training data». En: *Speech and Audio Processing, IEEE Transactions on* 13.3, págs. 345-354.
- Kenny, P., M. Mihoubi y P. Dumouchel (2003). «New MAP estimators for speaker recognition.» En: *Interspeech 2003*.
- Kenny, P. y col. (2008). «A study of interspeaker variability in speaker verification». En: *Audio, Speech, and Language Processing, IEEE Transactions on* 16.5, págs. 980-988.
- Kinnunen, T. y H. Li (2010). «An overview of text-independent speaker recognition: From features to supervectors». En: *Speech communication* 52.1, págs. 12-40.

- Kneser, Reinhard y Hermann Ney (1995). «Improved backing-off for m-gram language modeling». En: *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*. Vol. 1. IEEE, págs. 181-184.
- Kockmann, M., L. Ferrer y col. (2010). «Prosodic speaker verification using subspace multinomial models with intersession compensation». En: *Interspeech 2010*.
- Kohler, Mary A y M Kennedy (2002). «Language identification using shifted delta cepstra». En: *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*. Vol. 3. IEEE, págs. III-69.
- Kusner, Matt J y col. (2015). «From word embeddings to document distances». En: *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, págs. 957-966.
- Lai, S. y col. (2015). «How to Generate a Good Word Embedding?» En: *arXiv preprint arXiv:1507.05523*.
- Le, Ngoc-Tien y col. (2016). «Better Evaluation of ASR in Speech Translation Context Using Word Embeddings». En: *Interspeech 2016*.
- Le, Q. y T. Mikolov (2014). «Distributed representations of sentences and documents». En: *arXiv preprint arXiv:1405.4053*.
- Lebret, Rémi y Ronan Collobert (2013). «Word embeddings through hellinger PCA». En: *arXiv preprint arXiv:1312.5542*.
- Lebret, Rémi Philippe (2016). «Word Embeddings for Natural Language Processing». En: *infoscience*.
- Leeuwen, D. A. Van y N. Brummer (2006). «Channel-dependent GMM and Multi-class Logistic Regression models for language Recognition». En: *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, págs. 1 -8.
- Levy, Omer y Yoav Goldberg (2014). «Neural Word Embedding as Implicit Matrix Factorization». En: *Advances in neural information processing systems*, págs. 2177-2185.
- Li, Haizhou, Bin Ma y Kong Aik Lee (2013). «Spoken language recognition: from fundamentals to practice». En: *Proceedings of the IEEE 101.5*, págs. 1136-1159.
- Litman, D. J. y S. Silliman (2004). «ITSPoke: An intelligent tutoring spoken dialogue system». En: *Demonstration Papers at HLT-NAACL 2004*. Association for Computational Linguistics, págs. 5 -8.
- Liu, W.-W. y col. (2014). «Improved phonotactic language recognition based on RNN feature reconstruction». En: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, págs. 5322 -5326.
- López-Ludeña, V. y col. (2014). «Translating bus information into sign language for deaf people». En: *Engineering Applications of Artificial Intelligence 32*, págs. 258 -269.

- Lopez-Moreno, I. y col. (2014). «Automatic language identification using deep neural networks». En: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'14), Florence, Italy*.
- Lorenzo-Trueba, J. y col. (2013). «Towards Speaking Style Transplantation in Speech Synthesis». En: *Proceedings SSW8 2013-8th ISCA Speech Synthesis Workshop*.
- Lucey, S. y T. Chen (2003). «Improved speaker verification through probabilistic subspace adaptation.» En: *Interspeech 2003*.
- Manning, C. D. y H. Schütze (1999). *Foundations of statistical natural language processing*, págs. 277-279.
- Martin, A. F. y C. S. Greenberg (2010). «The 2009 NIST Language Recognition Evaluation.» En: *Odyssey*, pág. 30.
- Matejka, Pavel y col. (2014). «Neural network bottleneck features for language identification». En: *Proc. IEEE Odyssey*, págs. 299-304.
- McClelland, J. L., D. E. Rumelhart, R. G. Pdp y col. (1986). «Parallel distributed processing: Explorations in the microstructures of cognition, volume 2: Psychological and biological models». En: *MIT Press* 76, pág. 1555.
- Mikolov, T. (2012). «Statistical language models based on neural networks». Tesis doct. Ph. D. thesis, Brno University of Technology.
- Mikolov, T., W. t. Yih y G. Zweig (2013). «Linguistic Regularities in Continuous Space Word Representations.» En: *Hlt-naacl*, págs. 746 -751.
- Mikolov, T. y col. (2010). «Recurrent neural network based language model.» En: *Interspeech*, págs. 1045 -1048.
- Mikolov, T. y col. (2011). «RNNLM-Recurrent neural network language modeling toolkit». En: *Proc. Of the 2011 ASRU Workshop*, págs. 196 -201.
- Mikolov, T. y col. (2013a). «Representations of Words and Phrases and their Compositionality». En: *Hlt-naacl*, págs. 3111 -3119.
- Mikolov, T. y col. (2014). «Learning longer memory in recurrent neural networks». En: *arXiv preprint arXiv:1412.7753*.
- Mikolov, Tomas y col. (2013b). «Distributed representations of words and phrases and their compositionality». En: *Advances in neural information processing systems*, págs. 3111 -3119.
- Mikolov, Tomas y col. (2013c). «Efficient estimation of word representations in vector space». En: *in Proceedings of Workshop at ICLR*.
- Mogotsi, IC (2010). *Christopher d. manning, prabhakar raghavan, and hinrich schütze: Introduction to information retrieval*.

- Moreno, I. Lopez y col. (2014). «Automatic language identification using deep neural networks». En: *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, págs. 5337 -5341.
- Morin, Frederic y Yoshua Bengio (2005). «Hierarchical Probabilistic Neural Network Language Model.» En: *Aistats*. Vol. 5. Citeseer, págs. 246-252.
- Mulder, W. De, S. Bethard y M.-F. Moens (2015). «A survey on the application of recurrent neural networks to statistical language modeling». En: *Computer Speech & Language* 30.1, págs. 61 -98.
- Muñoz Igual, Ana (2015). «Evaluación y optimización de parámetros PLLR en sistemas de reconocimiento de idioma fonotácticos». En: TFG-ETSIT-Universidad Politécnica de Madrid.
- Muthusamy, Y. K., E. Barnard y R. A. Cole (1994). «Reviewing automatic language identification». En: *Signal Processing Magazine, IEEE* 11.4, págs. 33 -41.
- Nagarajan, T. y H. A. Murthy (2006). «Language identification using acoustic log-likelihoods of syllable-like units». En: *Speech communication* 48.8, págs. 913 -926.
- Ney, Hermann, Ute Essen y Reinhard Kneser (1994). «On structuring probabilistic dependences in stochastic language modelling». En: *Computer Speech & Language* 8.1, págs. 1-38.
- Penagarikano, M. y col. (2011). «Dimensionality Reduction for Using High-Order n-Grams in SVM-Based Phonotactic Language Recognition.» En: *Interspeech*, págs. 853 -856.
- Pennington, Jeffrey, Richard Socher y Christopher D Manning (2014). «Glove: Global Vectors for Word Representation.» En: *EMNLP*. Vol. 14, págs. 1532-43.
- Plchot, Oldrich y col. (2014). «PLLR Features in Language Recognition System for RATS». En: *Fifteenth Annual Conference of the International Speech Communication Association*.
- Povey, Daniel y col. (2010). «Subspace Gaussian mixture models for speech recognition». En: *IEEE*, págs. 4330-4333.
- Rabiner, Lawrence R y Biing-Hwang Juang (1993). «Fundamentals of speech recognition». En:
- Ramachandran, R. P., K. R. Farrell y R. J. Mammone (2002). «Speaker recognition general classifier approaches and data fusion methods». En: *Pattern Recognition* 35.12, págs. 2801 -2821.
- Renals, S. y S. King (2010). «Automatic Speech Recognition». En: *The Handbook of Phonetic Sciences, Second Edition*, págs. 804 -838.
- Reynolds, D. A., T. F. Quatieri y R. B. Dunn (2000). «Speaker verification using adapted Gaussian mixture models». En: *Digital signal processing* 10.1, págs. 19 -41.

- Reynolds, W. Campbell D. Sturim D. (2006). «Support Vector Machines Using GMM Supervectors for Speaker Verification». En: *IEEE Signal Processing Letters* 13.5, págs. 308 -311.
- Richardson, F. S., W. M. Campbell y P. A. Torres-Carrasquillo (2009). «Discriminative n-gram selection for dialect recognition.» En: *Interspeech*, págs. 192 -195.
- Richardson, Fred, Douglas Reynolds y Najim Dehak (2015). «Deep neural network approaches to speaker and language recognition». En: *IEEE Signal Processing Letters* 22.10, págs. 1671-1675.
- Richardson, M. (2009). «Principal component analysis». En: URL: <http://people.maths.ox.ac.uk/richardsonm/SignalProcPCA.pdf> (last access: 3.5. 2013). Aleš Hladnik Dr., Ass. Prof., Chair of Information and Graphic Arts Technology, Faculty of Natural Sciences and Engineering, University of Ljubljana, Slovenia ales.hladnik@ntf.uni-lj.si.
- Rodriguez-Fuentes, L. J. y col. (2016). «KALAKA-3: a database for the assessment of spoken language recognition technology on YouTube audios». En: *Language Resources and Evaluation* (2016), págs. 221 -243.
- Sagae, K. y col. (2009). «Towards natural language understanding of partial speech recognition results in dialogue systems». En: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*. Association for Computational Linguistics, págs. 53 -56.
- Salamea, C., L. F. D'Haro y R. Cordoba (2018). «Language Recognition Using Neural Phone Embeddings and RNNLMs». En: *IEEE Latin America Transactions, Aceptado y pendiente de publicación*.
- Salamea-Palacios, C. y col. (2013). «Incorporación de n-gramas discriminativos para mejorar un reconocedor de idioma fonotáctico basado en i-vectores». En: *Procesamiento del Lenguaje Natural* 51, págs. 145 -152.
- San-Segundo, Rubén y col. (2016). «I-Vector analysis for gait-based person identification using smartphone inertial signals». En: *Pervasive and Mobile Computing*.
- Saraclar, M. y B. Roark (2005). «Joint Discriminative Language Modeling and Utterance Classification.» En: *Icassp* (1), págs. 561 -564.
- Schwarz, P. (2009). «Phoneme recognition based on long temporal context». Tesis doct. Brno University of Technology, Faculty of Information Technology.
- Sebastiani, Fabrizio (2002). «Machine Learning in Automated Text Categorization». En: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval* 34.1, págs. 1-47.
- Singer, E. y col. (2003). «Acoustic, phonetic, and discriminative approaches to automatic language identification.» En: *Interspeech*.

- Song, Yan y col. (2013). «I-vector representation based on bottleneck features for language identification». En: *Electronics Letters* 49.24, págs. 1569-1570.
- Soufifar, M. (2014). «Subspace Modeling of Discrete Features for Language Recognition». En: *PhD thesis, Norwegian University of Science and Technology, Faculty of Information Technology*.
- Soufifar, M. y col. (2011). «IVector Approach to Phonotactic Language Recognition.» En: *Interspeech*, págs. 2913 -2916.
- Soutner, Daniel y Luděk Müller (2014). «Continuous Distributed Representations of Words as Input of LSTM Network Language Model». En: *International Conference on Text, Speech, and Dialogue*. Springer, págs. 150-157.
- Stolcke, A. y col. (2002). «SRILM-an extensible language modeling toolkit.» En: *Interspeech*, págs. 1-4.
- Stuhlsatz, André y col. (2011). «Deep neural networks for acoustic emotion recognition: raising the benchmarks». En: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, págs. 5688-5691.
- Sundermeyer, Martin, Ralf Schlüter y Hermann Ney (2012). «LSTM Neural Networks for Language Modeling.» En: *Interspeech*, págs. 194-197.
- Tellex, Stefanie y col. (2003). «Quantitative evaluation of passage retrieval algorithms for question answering». En: *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, págs. 41-47.
- Torres-Carrasquillo, P. y col. (2002). «Approaches to Language Identification using Gaussian Mixture Models and Shifted Delta Cepstral Features». En: *Lincoln Laboratory Journal, Massachusetts Institute of Technology*, pág. 4.
- Turian, J., L. Ratinov e Y. Bengio (2010). «Word Representations: A simple and general method for semisupervised learning». En: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, págs. 384-394.
- Vergin, R., D. O'shaughnessy y A. Farhat (1999). «Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition». En: *Speech and Audio Processing, IEEE Transactions on* 7.5, págs. 525 -532.
- Wang, Peng y col. (2016). «Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification». En: *Neurocomputing* 174, págs. 806-814.
- Wang, Yi y col. (2009). «Plda: Parallel latent dirichlet allocation for large-scale applications». En: *International Conference on Algorithmic Applications in Management*. Springer, págs. 301-314.
- Werbos, P. J. (1990). «Backpropagation through time: What it does and how to do it». En: *Proceedings of the IEEE* 78.10, págs. 1550 -1560.

- Weston, J., C. Watkins y col. (1999). «Support vector machines for multi-class pattern recognition.» En: *Esann*. Vol. 99, págs. 219 -224.
- Wiemer-Hastings, Peter, K Wiemer-Hastings y A Graesser (2004). «Latent semantic analysis». En: *Proceedings of the 16th international joint conference on Artificial intelligence*. Citeseer, págs. 1-14.
- Witten, I. H. y T. Bell (1991). «The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression». En: *Information Theory, IEEE Transactions on* 37.4, págs. 1085 -1094.
- Wong, E. y S. Sridharan (2003). «Fusion of output scores on language identification system». En: *Multilingual Speech and Language Processing*, págs. 1-5.
- Wu, L., S. C. H. Hoi y N. Yu (2010). «Semantics-preserving bag-of-words models and applications». En: *IEEE Transactions on Image Processing* 19.7, págs. 1908 -1920.
- Xu, Shunyi (2016). «Entity Linking with Convolutional Neural Network». Tesis doct. University of Calgary.
- Yan, Y., Etienne Barnard y Ronald A Cole (1996). «Development of an approach to language identification based on language-dependent phone recognition». En: *Computer Speech and Language* 10.1, págs. 37 -54.
- Yao, Kaisheng y col. (2013). «Recurrent neural networks for language understanding.» En: *INTERSPEECH*, págs. 2524-2528.
- Zaremba, W., I. Sutskever y O. Vinyals (2014). «Recurrent neural network regularization». En: *arXiv preprint arXiv:1409.2329*.
- Zhai, L.-F. y col. (2006). «Discriminatively trained language models using support vector machines for language identification». En: *Speaker and Language Recognition Workshop, 2006. IEEE Odyssey 2006: The*. IEEE, págs. 1 -6.
- Zhang, Xiang y Yann LeCun (2015). «Text understanding from scratch». En: *arXiv preprint arXiv:1502.01710*.
- Zissman, M. A. y col. (1996). «Comparison of four approaches to automatic language identification of telephone speech». En: *IEEE Transactions on Speech and Audio Processing* 4.1, pág. 31.